DOCUMENT RESUME

ED 247 892 IR 011 241

AUTHOR	Lewis, Matthew W.
TITLE	Improving SOLO's User-Interface: An Empirical Study of User Behavior and a Proposal for Cost Effectiveness Enhancements to SOLO. CAL Research Group Technical Report No. 7.
INSTITUTION	Open Univ., Walton, Bletchley, Bucks (England).
PUB DATE	21 Apr 80
NOTE	207p.
AVAILABLE FROM	Institute of Educational Technology, The Open University, Walton Hall, Milton Keynes, MK7 6AA, England.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC09 Plus Postage.
DESCRIPTORS	Cognitive Processes; College Students; *Computer Assisted Instruction; *Computer Software; *Design Requirements; *Error Patterns; Foreign Countries; Higher Education; *Programing; Programing Languages; Use Studies
IDENTIFIERS	Computer Users; Open University (Great Britain); *SOLO Programing Language; User Needs

ABSTRACT

This report describes an in-depth analysis of the errors made by users of SOLO, a programming language written for Open University students studying cognitive psychology. The study was designed to (1) determine the effectiveness of SOLO's current error-handling routines by evaluating how often SOLO produced "sensible" messages or automatic corrections; (2) examine the context surrounding errors and the kinds of conceptual misunderstandings which characterized their true causes; (3) suggest improvements to error-handling routines and other user aids; and (4) estimate the overheads incurred by such improvements and calculate the number of erroneous lines which could be successfully "trapped" or "pre-empted" by such improvements. An introduction describes the constraints from which SOLO evolved, features that "friendly systems" should incorporate, and features of SOLO. Data were collected by recording SOLO user interactions using the DEC-20 "Photo" program, which stores all the user input lines and SOLO response lines into a file in order of occurrence. Results are presented in relation to recommended changes in the system's user aids, specific type-related changes, and general changes. Appendices include a summary of all the changes, their requirements and benefits, and a breakdown of the errors that occurred by message and by types. (Author/LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

THE OPEN UNIVERSITY

ED247892

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Computer Assisted Learning Research Group

Improving SOLO's User-interface: an Empirical study of User Behaviour and a Proposal for Cost Effective Enhancements to SOLO

Matthew W. Lewis

CAL Research Group Technical Report No. 7

IR011241

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
E.S. Henderson
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Walton Hall,
Milton Keynes,
England MK7 6AA

Improving SOLO's User-interface:
an Empirical study of User Behavior
and a Proposal for Cost Effective Enhancements
to SOLO

by

Matthew W. Lewis

April 21, 1980

Contents

1) Introduction 1
1.1) Constraints from which SOLO Evolved. . 4
1.2) "Friendliness" Regarding Systems . . . 6
1.3) A General Review of SOLO Features. . .11
2) Description of the Data 15
3) Methods 17
4) Results 24
4.1) Changes to the User Aids 24
4.2) Specific Type Related Changes 45
4.3) General Changes 101
5) Discussion 110
6) References 114
Appendix A - Summary Pages 115
Appendix B - Breakdown by Types 124
Appendix C - Breakdown by Message . . . 145

IRO11241

Acknowlegments

The author wishes to thank Cathy Ramus for her help in preparing the manuscript, the Cognitive Studies group at Sussex University for their ideas, hospitality and facilities, members of the Open University Student Computing Service who put up with my questions, members of the O.U. Computer Assisted Learning group for their comments and especially to Marc Eisenstadt for his emotional, creative, good humored support and patience over the course of the project.

BEST COPY AVAILABLE

Summary

The main results of the analysis of the effectiveness of SOLOs user-interface are as follows:

--- Suggested changes could save an estimated 7.5-15.0% of the total correct time currently spent using SOLO. This part of the connect time is currently wasted on uninteresting, non-productive errors which could be handled more intelligently, or pre-empted completely by an improved systems. Many suggested improvements can be implemented trivially. While some would necessarily incur extra overheads in terms of CPU time, others would not, and the overall benefits appear to outweigh the costs.

--- The spelling corrector overall hit rate can be increased from 60% to 80%. Although it is currently well used and an integral part of the SOLO environment, it can be made more efficient.

--- The fine tuning capabilities of the spelling corrector (CAUTIOUS vs. TRUSTing mode), were never used. This type of finding points to the implementation of a revised, streamlined version of SOLO without the unneeded and possibly confusing addenda.

--- Using only the bad user input lines it is possible to diagnose many underlying conceptual problems. Once the problem is diagnosed, specific remedial teaching can be presented to remedy the situation.

--- Consistent user errors can point to flaws in the design itself. The method for accessing some of the HELP facilities proved to be the source of many user errors, hence the access method was changed to be more logical and consistent with the rest of the SOLO "machine".

My strong recommendation is to retain and improve the user aids and general environment. This is consistent with the goals of the university, that of being a leader in the field of distance teaching, and the desire to be truly cost effective. If SOLO's pedogogical effectiveness can be improved and connect time can be reduced by implementing changes, then the university stands to benefit twofold.

BEST COPY AVAILABLE

1) INTRODUCTION

SOLO is a programming language purpose written for Open University students studying Cognitive Psychology. It was developed specifically as an artificial intelligence (A.I.) language that would provide the tools for social science students with no previous programming background to carry out computer modelling of simple cognitive processes. The language is supported by a variety of user aids which are intended to make the user/system interaction as educationally sound and productive as possible.

The design of the language, user aids, help features and curriculum was carried out to produce an entire support structure for the SOLO user. I shall refer to this elaborate support structure as the user "environment". By correcting simple student errors and taking care of peripheral work (e.g., file handling, operating system interfacing, error message look up) the SOLO environment achieves an atmosphere of "friendliness". SOLO is well liked by D303 students but it is currently still far from being perfect.

Students still make trivial errors which to the human observer are mere 'slips of the pen' (or keyboard), but which generate confusing error messages. These messages are confusing be-

BEST COPY AVAILABLE

cause they don't necessarily address themsleves to what the student thought was happening inside the SOLO 'machine'. Thus, students can waste a lot of connect time (and CPU time) sorting out a fairly mundane problem. This wasted time could instead be spent on interesting conceptual issues of A.I., programming and cognitive modelling.

The catch is that this would require the provision of more "intelligent" user aids which could be costly in terms of CPU time. But just how "intelligent" would these aids have to be? The answer to these questions required an indepth analysis of the errors which SOLO users actually make.

This paper presents the results of just such an analysis, which was undertaken with four major objecives in mind:

(a) To determine the effectiveness of SOLO's current error-handling routines. This is done by looking at all the errors made by a group of SOLO users during a four week period of summer school use and evaluating how often SOLO produced 'sensible' messages or automatic corrections.

(b) To look a the context surrounding all of these errors and determine what kinds of conceptual misunderstandings characterize the true causes of these errors.

(c) To suggest improvements to SOLO's error-handling routines (and other user aids) so that they can deal appropriately

BEST COPY AVAILABLE

with the true causes of the errors as described in (b).

(d) To estimate the overheads incurred by such improvements, and to calculate the number of erroneous lines which could be successfully 'trapped' or 'pre-empted' by such improvements.

BEST COPY AVAILABLE

1.1) Constraints from which the SOLO environment evolved

When the D303 course unit was in its initial phases of design, the course team decided to include an emphasis on issues of artificial intelligence and the use of computer modelling to further the understanding of cognitive processes. Along with this emphasis, was felt the importance of having a "hands on" experience with A.I.; let the students get a feeling for the problems involved in the field for themselves. A programming language had to be chosen or developed that would meet the following constraints:

1) Introduce students to artificial intelligence, problems of representation of knowledge and cognition

2) Quickly assist students in learning high level programming constructs

   - Only two-weeks were alloted during the course for the entire SOLO project
   - Access to terminals was projected to be only a maximum of 10 hours of connect time during regular course.

3) The system must be extremely "friendly"

   - there would be no available on-site tutor during the terminal sessions
   - students do the preparation and programming while working alone at home
   - students would have limited typing skill

4) The system would have to be suitable for social science students who have non or anti-computing backgrounds.

5) The language would have to cover fundamentals of programming, (i.e. algorithm, flow of control, conditional branching, structured programming, etc.)

BEST COPY AVAILABLE

6) The computer use must be scheduled for the beginning of the term when other computing classes are at a minimum usage of the O.U. DECsystem-20.

These specific constraints under which the course team was working were combined with the general constraints by which any beginners programming language that hopes to be successful is constrained. The main points here are:

1) Students should be able to do exciting things on the first day of the computing experience to initially catch the interest of the learner.

2) The language should have direct and obvious relevance to the students' main interests at the time (in our case psychology or linguistics).

3) The overall experience should be enjoyale or pleasurable to leave the user in a positive state of mind about the system, language and his/her abilities.

Since no existing language met these requirements, a language and environment were purpose designed, called SOLO. This name was meant to emphasize the nature of the language; that it was designed to be learned and used by people working on their own. SOLO is a specially tailored, non-numerical programming language written by Marc Eisenstadt of the Open University Psychology faculty. The language was implemented by Vince Cook, Dave Findel- Hawkins and Mark Woodman of the Open University Student Computing Service on the DECsystem 20. After a three month period of piloting on a group of test students, final changes were made and SOLO was ready to go into service with the D303 course when it was introduced in February 1978.

BEST COPY AVAILABLE

SOLO is now in its third year of operation, being used by approximately 400 students annually. The general student reaction to SOLO, as guaged by interviews at summer schools, has been encouraging. Not only do students seem to enjoy the SOLO work, but the student projects and tutor marked assignments have demonstrated that SOLO has been successful, to a large extent, in achieving its goals while staying within its constraints. Students appear to be picking up the high level programming concepts as well as meeting and exploring problems of representation and modelling, all within the allotted two weeks of course time.

1.2) "Friendliness" regarding systems, and how one incorporates it into a learning situation.

The following points are a non-exhaustive set of ideals which "friendly" design should incorporate. A brief, informal statement of SOLO's rating on each point is included. These statements are expanded and explained later in the paper.

1. A general definition of friendliness proposed here is; 'the features of the operating system which cut down on the amount of overhead involved in using the system', (e.g., decreasing the number of characters that must be typed to achieve an operating system call, or how much manual referencing is needed to perform fancy, seldom used commands.) In simpler terms, a friendly system is one which makes it as easy as possible for the

BEST COPY AVAILABLE

user to get the system to do what she/he wants it to do.

A concrete example of friendliness in a learning situation can be seen in the interaction of a good human tutor with a student. Take, for example, tutoring a young child in arithmetic who has a problem with writing the number "3" backwards. If the child makes this mistake, the friendly tutor does not make the child rewrite the entire problem. Instead, the tutor might point out the error, help with the correction and warn the student about the necisasity for clarity in his/her writing and continue with the problem. The overall goal is to expediate the process of learning to do arithmetic. The tutor knows what the student wanted to do and used that knowledge to help achieve the learning goal while at the same time keeping the number of careless errors to a minimum.

This philosophy is borrowed from Teitelman (1974) and his formalization of "Do what I mean" (DWIM). A system operating under the DWIM philosophy helps to intelligently trap careless or unimportant errors and hence leads to optimizing productive interactions with the operating system.

It should be stressed here that the author is not advocating the position that students should not be allowed to make errors. On the contrary, errors are an important part of the learning process. Instead of cutting out all the errors, a friendly sys-

BEST COPY AVAILABLE

tem should attempt to ensure that of the errors made by the user/student, interesting conceptual errors should be maximized, and time consuming simple, "obvious" errors wich are not productive to the conceptual growth of the user, should be minimized.

SOLO rates relatively well in this respect. Its spelling correction and explicit prompting for flow and control pre-empt many such errors, however there is much room for improvement.

2. This DWIM philosophy can also be taken too far. As mentioned earlier, the system should do what you mean, but not at the expense of learning concepts. One can imagine a hypothetical computer tutor which knows the domain and answers to problems which the learner is attempting to solve. If this tutor was not sensitive, and too powerful it could conceiveably simply pull the learner through the problem solving steps towards the solution without any learning taking place. For example, if the problem were to write a program to do a specific task, the tutor program might, taken to an extreme, actually systhesize the answer for the student. By doing so, the student has been denied the conceptual work and subsequent learning. Currently SOLO can be modified to increase its power as a learning tool and still remain in the "safe" realm of DWIM philosophy.

3. The system should maintain as consistent a conceptual model of itself as possible for the user. This includes not in-

BEST COPY AVAILABLE

creasing the possible friendliness of the environment if this entails adding rules or kluges which don't fit into the existing conceptual model and rule system with which the user is working. An example of this would be correcting spelling errors without a prompt or message to the student. This could conceivably introduce the idea that the student can be careless speller in all cases, and possibly even generalise this rule to syntax as well. SOLO rates high in this respect. Pains were taken to keep the environment conceptually consistent, yet exceptions to this have been pinpointed by the analysis and are areas of recommended change.

4. Friendly learning systems should have multiple approaches to teach a single concept. As all teachers know, the best written lesson won't reach all the learners. Since not all people learn the same way, a good human tutor has an assortment of models and examples to teach a single concept. An intelligent system should have the same "bag of tricks" if it is to be maximally effective. By having multiple approaches to a concept the probability of reaching more students is increased. This "shot gun" effect is especially important for learners working alone. SOLO does not rate well here. Single conceptual models are used and are relatively well developed. This, however, does not substitute fully for the power of multiple approaches.

5. A friendly system should be user or self-adjusting to fit

BEST COPY AVAILABLE

the growth in knowledge of the user from beginner to expert. As the person gains knowledge or demonstrates some threshold of expertise, options should become available which increase the power of the system. These options should be passively available to, but not forced on the beginner. Although some people want to know all the available features of a system from the outset, there is also a group who would tend to undergo cognitive overload if presented with all the features at the start of the computing experience. Ultimately, as the person developed a need, or saw a possible way to save time using the operating system they should be able either to access a file of options to meet their need, or given the tools to define their own macros of operating system commands. This method of personal discovery is desirable for some learners, but a more assertive method is needed for people who wouldn't think to invent new commants, or to whom certain ideas for time saving cammands simply didn't occur in one or two case. Here, SOLO can be improved. Currently it has two modes, but could use an extra helpful

6. Friendly systems should have multiple sources or depths of on line and manual help information. This is desirable for two reasons. First, it decreases the possiblility of a user getting into a situation where he/she has run out of alternative error handling schemes. This is wrapped up in the fear of many beginning programmers of simply getting stuck somewhere in an operating system, either in a strange mode, or staring at an in-

BEST COPY AVAILABLE

comprhensibie error message and running out of places to turn to for help. In addition, beginners should be forced to familiarize themselves with these help option. A general observation about humans is that we are more likely to try something that we have tried before and had success at, than something new or novel (espescially in a threatening situation).

Second, it gives the option of choice in the amount of information the user wishes to receive. An experienced user might only need a quick reminder about syntax whereas a beginner might need a paragraph or manual reference to solve a systax question. SOLO has the framework for good on line help information, but it is currently under developed. Changes are suggested.

1.3) A General Review of Existing Solo Features.

As all programming languages, SOLO has a set of runtime and compile time error messages. These messages are generally well thought out and in some cases include a reminder about getting additional information about the current error from the "HELP" call. The "HELP" call ca be carried out either for the current error, or to get information about specific systax problems. In the first case the user types only "HELP", and in the later "HELP <statement name>". The message in response to these requests gives more specific detail about the error culprit, as well as possble causes and solutions to the problem. A reference to the

BEST COPY AVAILABLE

SOLO primer page is included as an additional source of assistance if necessary. However there are cases where these messages are vague, misleading or simply don't apply to the error made.

Also included in SOLO's design are sophisticated prompting and indenting features for conditional statements. These features are meant to pre-empt flow of control errors by forcing an explicit decision about flow of control at each branch of a conditional statement. Beginning users often appear not to consider both cases of their simple control flow statements. This feature also acts as a convenient debugging message feature by essentially commenting the step and presenting a procedure slot to fill if desired. This procedure slot is well suited for simple user generated debugging messages. In addition, syntax is explicitly demanded and the required word possibilities presented to the user if they have been forgotten. Not only does this cut down on flow of control errors, but it reduces memory load on the new learner. S/he can rely on a reminder from the system if faulty syntax is attempted. This appears to be a relatively robust feature of SOLO, giving in most cases, much needed assistance. There are also some flaws that will be the target of several recommendations for change.

The "Transparent machine" philosophy, around which SOLO was designed, is based on the work of Du Boulay and O'Shea (1980). This philosophy basically states that all the effects of user and

BEST COPY AVAILABLE

program interaction with the environment are made explicit to the user. By specifying explicitly such things as changes to the data base, the unknown or "black box" quality of the operating system is reduced. This is desirable in that if there are no hidden or "magic" effects or side effects, then more complete understandings of, and confidence in SOLO use and control can be established. Flaws and improvements in SOLO's transparency, as pointed out by consistent user errors, will be detailed later.

The SOLO environment also has a set of user controlled "HELP OPTIONS" which are set with a default values aimed at the beginning student. A "+" indicates that this option is in effect from the default setting. The first options are related to the spelling correction feature with which SOLO is equipped.

SPELL (+) Enables any misspellings of procedure names to be automatically corrected.

NOSPELL (-) Completely disables spelling corrector, thereby also aliminating the next 4 options shown.

TRUSTING(+) Performs most corrections without asking approval.

CAUTIOUS(-) Requests approval before preforming corrections.

NODEOFF (-) No spelling correction on nodes & relation names.

NODEON (+) Applies spelling correction to nodes & relation names.

Next are options related to the printout that the user receives:

BEST COPY AVAILABLE

REACT (+) Describes nodes whenever NOTE & FORGET are used.

SILENCE (-) Supresses description of node after NOTE and FORGET.

TRACE (-) Show each step number as it is carried out.

NOTRACE (+) Normal execution (no step numbers shown).

BRIEF (-) Short error messages (particularly 'Wrong format').

VERBOSE (+) Normal error messages.

Options related to 'undoing' things that the student has done:

RESTORE (-) Restores data base to what it looked like when the student logged in most recently, but leaves the procedures intact.

START (-) Like RESTORE, but also erases the definitions of all procedures that were defined during the current session (older procedures and older changes to the data base are left alone).

These HELP OPTIONS are examined in section 4.1 where evaluations and suggested changes are made.

BEST COPY AVAILABLE

BEST COPY AVAILABLE

2) Description of the Data

The sample of data used in this analysis is taken from actual interactions of 1979 D303 summer school students with SOLO. One of the options of D303 summer school is to carry out an A.I. project using SOLO. Students spend three days working in groups of three or four constructing a model of some cognitive process. Some problems are presented by the tutors, but the students are free to pick a problem for themselves if they wish to do so. Since this is the situation in which the data was gathered, it is critical to point out that this is not a random sample of beginning SOLO user interactions. The people who were generating the data were self-selected to work on the A.I. option. Tutor reports indicate that people who choose to work on this project are generally from two groups:

One group consists of people who enjoyed the SOLO work during the course and desired to do more. This group sometimes includes a subgroup who are experienced in some other programming language. People in this subgroup are encouraged to take another option that might prove to be more challenging for them.

The other major group of A.I. project selectors are people who didn't understand the SOLO project to their satisfaction during the course, and thus wanted additional SOLO experience. The common denominator of the people in both groups that seeems to set them apart from the average SOLO user is a high level of

motivation to learn to write working SOLO programs.

It is also worth pointing out that these people had all spent 1-2 weeks using SOLO, about 3 months before attending summer school and thus had time to think about the language and principles of A.I. In addition these users are not working alone. In fact, they have ready access to tutor and peer assistance at all times.

These biases are pointed out to stress that the current analysis was done on data gathered under optimal learning conditions. Another study of actual beginners' interactions with SOLO is already under way and may lead to a whole new analysis of errors and subsequent suggested changes to SOLO.

BEST COPY AVAILABLE

3) METHODS

The SOLO/user interactions were recorded using the DEC-20 "Photo" program. This program simply stores all the user input lines and SOLO response lines into a file, in order of occurrence. This kind of data file is reffered to as a "dribble file", connoting the slow "dribble" of information into a storage space. This label is borrowed from the work of Feurzeig (1969), who first coined the term.

My program (written in PASCAL) took these dribble files as input, and produced as output seperate files, called, message indices, containing all of the occurrences of each individual SOLO message and "annotated" dribble files (to be explained shortly). For example, every occurrence of message number 12 ("Oops... you forgot the line number"), along with the offending line and immediate context which produced it, was sorted into message index number 12. An "annotated" dribble file is simply a dribble file to which line numbers have been added and error messages flagged in the margin at each message occurrence. In the event of needing more context to disambiguate the cause of an error, these features make the task of manually re-examining the dribble files far easier and more direct.

The message indices were examined in detail in order to discover if there were general types of input which were receiving a

single message. For example, a single wrong format-error message is generated as the response to two distinct types of errors: (a) too many arguments passed to a procedure, and (b) too few arguments passed to a procedure. Each of these types have specific subgroup types which arise from specific causes. These range from forgetting to add a space between arguments, to serious conceptual misunderstandings. It is interesting to break down the occurreces into types such as this because these different types might be more intelligently handled as seperate cases with seperate messages, or some types might be pre-empted all together if an automatic correction could be easily generated.

Type descriptions have the following format, using type 1 as an example:

t1) Too many arguments to a procedure

e.g. NOTE FIDO LIKES WEST HAM UNITED

number of occurences: 52

syntactic category: too many args

conceptual category: unlimited lists possible

status: improve message

priority: low RTC:2 EST: 05:40

--The first line of the type description contains the type number (t1 in this case), and a brief description of the type of error.

--The second line contains an example of the occurrence of this type of input.

--The third line contains the number of times that this type occured in the message index.

--The fourth line contains a brief statement of the general syntactic category into which the error could be classed. This classification was added in an attempt to make the results of this specific study of SOLO generalize to other programming languages in which similar error types might occur.

--The fifth line contains the type of conceptual misunderstanding which could have produced the error. Again, an attempt was made at making the descriptions of these categories as general as possible in order to facilitate comparisons with other languages.

--The sixth line contains the status of the type. "Status" falls into three main categories;

ok :the type is well handled by the current implementation

improve/new message: the type would be better handled if the message were changed or an entirely new message was given

trapable: some or all of the occurrences could be pre-empted so that they need not occur at all

--The seventh and final line gives the priority for the change. Priorities may be any of the following:

high: the change is either very important for eliminating students' confusion, or moderately important and involves a large number of occurrences

medium: the change might only be important for a few occurrences, or is slightly important, and involves a large number of occurrences

low: the change isn't of critical importance; it might correct a few minor points, or handle only a small number of very specific errors.

na: this stands for not applicable and occurs only when the status is set at "ok", hence there is no change.

This final line also contains some information regarding recovery time estimates and classification for the error type. These are explained in detail below.

BEST COPY AVAILABLE

Without the record of elapsed time between user inputs to SOLO, it is impossible to make an accurate estimate of the time savings which would be gained by an intelligent user aid which pre-empted certain types of errors. The Photo program which produces the Gribble files should be augmented to include a record of the time at which lines were input. Until this information is available the probable time savings can only be roughly estimated.

The "guesstimation" used to approximate the amount of connect time saved for each change was based on the division of types into 5 recovery time classes (RTCs). These classes characterize the amount of time a student curretly wastes on errors which are tangential to his/her main programming activities, and thus the amount of time which would be saved if these errors were automatically corrected, or at least had a totally unambiguous error message:

RTC 1) No time saved: changes that don't help the student appreciably but add to the consistency of the environment or are purely cosmetic

RTC 2) Easily found and corrected types: 10-30 seconds to recover from error

RTC 3) Possibly confusing types: 30 sec.-5 min. to recover from error

RTC 4) Complicated types: 5-15 min to recover from error

RTC 5) Really nasty or invisible types: 15 min. to recover, or fatal errors

"Recovery" here is meant to include all the time it takes to correct the error made. This includes getting the error message, figuring out the meaning of the message, identifying and listing the procedure (if necessary), finding the occurrence of the actual bad bit of code and retyping it (re-entering and terminating EDIT mode if necessary). The variation in recovery time between students is great, however. Typing slowness alone can double or triple the recovery time for a "hunt and peck" typist. The times given are the estimated average difference between recovery before and after the proposed change.

"Estimated saved time" (EST) is calculated by multiplying the number of occurrences of a type by the minimum of the time span estimated to recover from the error. For example, type 27 occurred 13 times. It was given a recovery class rating of RTC: 3 (30 sec.-5 min.), hence the time saved by trapping the error and pre-empting the occurrences would be:

EST = 13 * 30 sec. = 7.5 minutes of connect time

Times were computed for all proposed changes and are summarized in the "Summary Pages" in appendix A of this paper. The types referred to earlier are collected, with the appropriate breakdowns by message number, and summarized on "Message Analysis Sheets" which are included in appendix B.

The type numbers (designated by the letter "t"), were as-

BEST COPY AVAILABLE

signed as the types arose from the data, and are a separate numbering system from the message numbers. The latter are designated by the letter "m" and are the numbers which SOLO uses internally to access the messages. The recommended changes are specific to each type, e.i., change 27.1 reflects what was learned from type 27. Again, a list of all the types is included in Appendix B, along with a summary sheet which references message numbers by the type number. For example,

Type 1: Too many arguments to a procedure	Fraction of total in message index
m18: the format for NOTE is:	19/156 *
m19: the format for DESCRIBE is:	12/24
m20: the format for FORGET is:	1/42
m22: the format for LIST is:	2/24
m23: the format for EDIT is:	2/22
m25: the format for CHECK is:	3/44
m43: Wrong format	7/9
m101: type options here:	6/50
Total number of occurrences of this type:	52

* This means that message m18 ('the format for NOTE is...') occurred a total of 156 times in the entire collection of dribble files. Out of these 156, only 19 were due to error type no. 1 ('Too many arguments to a procedure'). The other 137 occurences had different underlying causes which are cross referenced in the appendix C.

The decision of long or short term status for a change was

contingent upon several factors. If the change required the addition of a special demon, major restructuring which would require a rewrite of the manual, or introduction of a data type, then the change was designated as "long term". If the change required the addition or changing of a canned message, the turning off of an existing feature, or a simple change to a format, then it was classed as short term change. These classifications are indexed on the Summary Pages located in appendix A.

BEST COPY AVAILABLE

4) Results

The results have been broken up into three sections. Section 4.1 is the summary of the analysis and subsequent changes to the user aids specifically. Section 4.2 covers the changes which were designed to handle specific user error types generated during normal top-level SOLO use. Section 4.3 is a collection of general changes which are not direct products of the empirical results of this study. Instead, these are errors or design flaws which have been noticed during informal tutoring sessions or personal use by members of the course team, specifically Marc Eisenstadt. The results follow:

4.1) Changes to the User Aids

The analysis of the dribble files revealed that error handling was not the only way in which SOLO could be improved. Other user aids, such as the TRACE facilities and the design of the spelling corrector, were causing difficulty.This difficulty centered around (a) access to the user aids (e.g. invoking TRACE was hard to do), and (b) performance of the user aid (e.g. the spelling corrector algorithm is too dumb).

BEST COPY AVAILABLE

The categories of changes to the user aids are shown below:

4.1a Restructure and improve the spelling corrector

4.1b Change the user access to the 'HELP OPTIONS'

4.1c Improve the tracing facilities

4.1d Review remaining options

4.1e Rename some options

4.1f Possibly add a "very helpful", verbose beginners mode

4.1g Add a lesson in HELP usage

These suggestions will be elaborated upon, and their drawbacks and benefits discussed. Increased understanding and time savings to the user, as well as other related factors will be presented.

4.1a) Restructure and improve the spelling corrector

The current structure of the spelling corrector features is ambiguous and for the most part the available features are unused. The options to turn the corrector on and off (no fine tuning) were used 18 times out of a total of 389 uses of the different OPTIONS, and 16 of those were to turn it off. This is most probably due to the fact that the spelling corrector in its current form is not very efficient (see change 13.1). The options to fine tune the corrector, namely;

TRUSTING (does not require user confirmation)
CAUTIOUS (asks for user confirmation)
NODEON (performs correction on node and relation names)

BEST COPY AVAILABLE

NODEOFF (no correction on node and relation names)

were never used.

For these reasons, it is recommended that the spelling correction aid setting be changed to simply SPELL and NOSPELL with a constant TRUSTING mode setting for procedure names, and a constant CAUTIOUS mode setting for node and relation names. The options mentioned above would be removed. This change is only recommended in conjuntion with the implementation of an improved spelling corrector algorithm, which is detailed later in this section. Before the discussion of the algorithm, the structure of the spelling correction feature will be explained.

The corrector is broken into two components. One component is the procedure-name spelling-corrector, which makes changes for the user without asking for confirmation on the change. The second component is the node-name spelling corrector which asks the user for confirmation before making a change, e.g.,

When you typed JOAN did you mean JEAN (yes or no)?:

This component gives the user veto power on proposed changes because there tend to be many more 'near-misses' involving node-names than there are with procedure-names.

Analyses were performed on both components. The results of the procedure-name spelling corrector (forced correction) are presented below followed by a discussion.

BEST COPY AVAILABLE

Procedure-name Spelling Corrector Results

Category type	Example	Number of occurences
Correct change	10PRINT = 10 PRINT	271
Wrong split	POSSIBLY HAS = PO SSIBLY HAS 10Assign = 10A ssign	8
Split line number	2A FORGET = 2 A FORGET	21
Synonym Finder		
--Correct	DELETE = FORGET	2
--Steam-rolled	COMPARE = CHECK	9

number of times correct	273/311 (88%)
number of times incorrect	38/311 (12%)
proportion of incorrect occurrences that proposed changes will pre-empt	30/38 (10%)
Overall hit rate:	
----Before proposed changes	273/311 (88%)
----After proposed changes	271/279* (97%)

* This is 279 rather than 311 because the split line number and synonyms will no longer occur. Hence, the total number of checks goes down.

The category called "correct change" contains all the occcurrenes of spelling correction where SOLO made the proper decision about a user error. The "Wrong split" category consists of instances where SOLO added unwanted spaces to a user input line causing an error to occur. These occurrences can be pre-

BEST COPY AVAILABLE

empted in the following way: if the proposed 'split' results in either a procedure name which does not yet exist (e.g. 10 Assign = 10A ssign) or a procedure which already exists but has a different format from the one posited by the split, then the user is simply asked to confirm the correction before it is carried out.

The third type presented above, that of splitting line numbers, would be completely pre-empted by the implementation of a line number demon (see change 75.1). This demon would recognise the illegal line number and give an intelligent message about correct the use of sublines.

The final category of synonym correction is covered in change 13.1. The general recommendation is to turn off the synonym finder because it appears to be doing more harm than good, (note: this change has already been implemented).

In sum, the spelling procedure name corrector appears to be doing a relatively good job currently, but can use improvement. With the proposed changes operating, the performance would go to be 271 correct applications of the spelling corrector, and only 2 missed cases where the corrector would miss a possible synonym change.

Next, the node-name spelling corrector results will be presented and discussed. The results are as follows:

BEST COPY AVAILABLE

Node-name Spelling Corrector Results

Category Type	Example	No. of Occur.
	"when you typed..."	
Plausible correction (word +/- 2 charactors)		
Accepted	"MAYHAVE did you mean MAY HAVE"	130
Rejected	"JEAN did you mean JOHN"	49
Silly number check	"14 did you mean 1"	
Accepted		2
Rejected		93
Ridiculous Possible match	"JUMPROPE DID YOU MEAN J"	
Accepted		4
Rejected		93
total		371
number times plausible		179/371 (48%)
number times pre-emptible		186/371 (50%)
number times missed by proposed changes		6/371 (2%)
hit rate		
---before proposed changes		136/371 (37%)
---after proposed changes		130/179 (73%)

The first category listed above, "Plausible correction",is

BEST COPY AVAILABLE

BEST COPY AVAILABLE

defined as checking an input word against another word only if the words differ by 2 or less characters. This type was formed from the observation that the SOLO node-name spelling corrector was doing a lot of ridiculous possible matches, most of which were being rejected and only causing time wastage and type ahead errors. With the rule of +/- 2 character differences between words as a guide, the hit-rate improves drastically.

The "Silly number check" category proves to be an area for great improvement in the decision algorithm of whether to check for a spelling correction or not. This is most true for the summer session where the subtraction problem is attempted by many students and are used numbers extensively as nodes and relation names. It is proposed that a hidden data type be introduced which adds a distinction between numerals and letters. This change would also come into play with the line number demon (see change 75.1) used in the spelling corrector and the editor. If 2 nodes/relations differed by only one or two numerals then they should not be corrected. This would improve the corrector by cutting out these specific numeral use checks when attempting the subtraction problem, but it would also relieve the user of useless checks like:

"when you typed PRAISE1 did you mean PRAISE?"

This is a common naming algorithm for variations on procedures, used by beginners and summer school users alike, and the proposed corrections of this type were never accepted.

The final category of "Ridiculous possible matches" could be pre-empted. The occurrences in this category were all checks of input words with possible existing words, but the words differed by more than two characters. The spelling check here could be contingent upon the argument counting (see change 1.1). If an argument number error were found by the demon, it could guide the spelling corrector to either crunch pairs together (in the case of too many arguments) or make logical splits of nodes or relation if there were too few arguments, e.g.,

SOLO: NOTE FIDO ISADOG

The argument counter knows that there are too few argument, so it looks for a likely split. The spelling corrector recognises the two node names, (or a single node name) and offers a split.

when you typed ISADOG did you mean ISA DOG

This case occurs only when the spell corrector has failed to add the space.

Similarly, 'pair-crunching' could be attempted in the following case:

SOLO: NOTE FIDO IS A DOG

The argument counter cues the spelling corrector that too many arguments exits here, and a possible crunch should be attempted. The spelling corrector then attempts to remove spaces between the arguments in an attempt to match a pair of the arguments to a

BEST COPY AVAILABLE

single argument that had already been defined. e.g.,

When you typed IS A did you mean ISA ?

In the revised implementation this change would pre-empt 44 format errors due to careless unwanted spaces. This change specification is reiterated, and expanded upon in change 3.1.

These proposed changes in the node-name spelling corrector improve the performance immensely by saving 186 unecessary requests for confirmation and pre-empting 44 possible format errors. These tightening measures on the algorithm would result in only 6 cases of desired spelling changes <u>failing</u> to be presented. This trade-off seems worth the time savings however, and, in addition, it is not known whether the responses in these 6 cases were wise choices based on the message at hand, or simply random responses by the user.

The overall performance of the combined spelling correctors improves from a combined hit rate of 60% with the current implementation to a hit rate of 88% with the proposed changes. This savings in unnecessary disturbances to the user and added connect time due to waiting for output give a strong reccomendation for these changes to be implemented.

4.1b) Change User Access to the 'HELP OPTIONS

The current access to the HELP OPTIONS feature of SOLO is

BEST COPY AVAILABLE

achieved by first getting into a special mode which presents the options available, then forces the user to select an option. Currently, to get into this special mode, the user types:

SOLO: HELP OPTIONS

The HELP OPTIONS are listed and a forced decision is given to select an option. Following the forced decision, the user is given a confirmation statement to decide if any further option selection is desired, e.g.

```
SOLO: HELP OPTIONS
    The options available are ....
Type option here: TRACE
Do you want to select any more options? (Yes or No):
```

This design proves to be relatively clumsy, in several ways. To begin with, it is a forced decision with no alternative of simply returning to the top level of SOLO. Secondly, the confirmation statement was positively responded to 129 times, and negatively responed to 388 times, (see change 50.1). This implies that perhaps a negative default setting would be advisable for this option. Type ahead problems only complicate matters for this confirmation message, with 19 occurrences of people forgetting that they had to make the yes/no decision before going back to top level SOLO. In addition, students have attempted many different ways to speed up their access to these options. These methods have, on the whole, been logical, consistent ideas for

BEST COPY AVAILABLE

possible ways to restructure the feature selection. For example:

SOLO: HELP CAT

SOLO: HELP OPTIONS TRACE

SOLO: TRACE

These errors, along with the fact that there are several occurrences of people getting "caught" in HELP OPTIONS and typing "DONE" or "EXIT" (in an apparent attempt to get back to top level; see change 3b.1), point to the need for restructuring the HELP OPTIONS feature.

The recommended restructuring involves two changes:

(a) Change from the name 'OPTIONS' to the name 'AIDS', to reflect the concept of a set of tools helpful to the user. To access information about the aids the user would typed:

SOLO: HELP AIDS

The response to this command would list the aids and their functions (and current settings, whether on or off), give brief directions for use, then return to top level.

(b) Direct access to the user-aid calls from top level (or indeed, from within user defined procedures). The two reserved words "ENABLE" and "DISABLE" would be the invocation functions for the chosen aid.

When an option is activated by means of this top level call,

BEST COPY AVAILABLE

SOLO should give a confirmation that the desired effect has been achieved. A short message, stating the previous state of the option, and the current, updated state of the option would be desirable, e.g.,

```
SOLO:  ENABLE TRACE
OK...
was    DISABLE TRACE
now is ENABLE TRACE
SOLO:
```

"HELP" is maintained as a device for getting information from the system, and new invocation functions are added to reinforce the distinction that using the aids is a different level action than using the normal SOLO procedures. The special access of a reserved word is required to alter the SOLO environment (i.e. user aid) itself.

One advantage of this restructuring is that many of the spelling errors which occurred in the old structure would be eliminated. This is due to the fact that a large proportion of these errors were caused by the inclusion of an "O onto the begininng of the desired option name, e.g.,

```
Type Option Here: OSILENCE
```

This was most probably due to people attempting to type control O to suppress the output and instead typed capital O. The new structure would have no prior output to suppress, hence the lack

BEST COPY AVAILABLE

of this very specific error.

The current tracing facilities of SOLO are relatively will used (98 times turned on or off out of a total of 389 manipulation of all the options combined), but they are severely lacking. Much improvement could be gained by the addition of some extra information to the existing trace.

Currently, TRACE provides only the bare minimum of useful inforamation to help guide the user to the source of his/her error or bug. When a procedure is run, the procedure name and the line numbers are printed as they are executed. Any output that occursis printed after the line number of the line that generated it. If a sub-procedure is called, its name and executed line numbers are printed, and a slight indentation is added to the output while in the sub-procedure. Again, the basic design is acceptable, but improvements are needed.

To begin with, the indentation which acts as a demarcation for procedures should be more distinct. A larger indentation, filled with periods in groups of 3 or 5 would be useful in displaying more graphically the flow of control from procedure to procedure. This indent would be added at every point where a procedure call was made. The values of any parameters passed at that call would also be stated. For example, suppose the procedure FOO was being traced, and that it had the following definition.

BEST COPY AVAILABLE

```
TO LIKECHECK /X/
1 CHECK /X/ LIKES ?
....1A If present: SUB1 * /X/; CONTINUE
....1B If absent: ;EXIT
2 PRINT "THAT'S ALL FOLKS"
DONE
```

Suppose that SUB1 had the following definition:

```
TO SUB1 /FIRSTPERSON/ /SECONDPERSON/
1 CHECK /SECONDPERSON/ LIKES /FIRSTPERSON/
...1A If present: PRINT "SOUNDS LIKE A GOOD THING";EXIT
...1B If absent: PRINT "THAT'S TOO BAD"; EXIT
DONE
```

If the database contained only this:

```
JOHN---LIKES--->MARY
```

and we typed "ENABLE TRACE" and then invoked LIKECHECK with the parameter value of "JOHN", the TRACE would appear as follows (my comments are given in square brackets):

```
SOLO: LIKECHECK JOHN
LIKECHECK JOHN
LIKECHECK(1) * = MARY
LIKECHECK(1A)
        [ line 1A of LIKECHECK now calls procedure SUB1
            with values /JOHN/ and /MARY/ ]
..... SUB1 JOHN MARY
..... SUB1(1)
```

Results Section 4.1: Changes to the User Aids

BEST COPY AVAILABLE

```
..... SUB1(1B)
THAT'S TOO BAD
        [ procedure SUB1 has exited, control is being passed
                back to procedure LIKECHECK ]
LIKECHECK(2)
THAT'S ALL FOLKS
DONE
```

Printing the values of any variables or parameters that are called or passed is also of critical importance. The example above prints the value of * in step 1 of LIKECHECK. This increased explicitness would save the addition of print statements for use in debugging. It is recommended that the procedure name be included with every line number of the procedure. In the case of multiple nested subprocedures it is very difficult to keep track of which procedure is which. This format would be as follows:

<Procedure name>(<line number>)<Parameter/variable values on line>

With the addition of these changes, the TRACE facility would become a very useful debugging tool, even for the beginner. The use of TRACE as a learning tool, i.e., a way to see the flow of

BEST COPY AVAILABLE

control explicitly, would also be a great benefit.

4.1d) Review of Remaining Options

Having covered the TRACE and SPELL facilities, the remaining options will be reviewed.

---The SILENCE feature which suppresses description of nodes after NOTE and FORGET, was selected 111 times. The negation of the SILENCE option, REACT was selected 6 times. SILENCE should be changed so that it does not suppress the result of a top-level check statement as it does in the current implementation.

----The BRIEF mode was selected 5 times and reset to VERBOSE only once. This feature appears to be acceptable, if the current structure is maintained. Only the addition of an alternate access word (i.e., NOBRIEF) would be desirable (explained below).

----The three options related to restoration of the data base were used as follows:

RESTORE - restores data base to pre-login state 2
START - restores procedures and data base to pre-login state 0
REVERT - restores data base to 'virgin' state (1st user session) 16

and are performing adequately.

BEST COPY AVAILABLE

----The most frequently used option was the CAT option which lists the users complete data base and procedure list. CAT was used 132 times, and the only change which would be appropriate for this feature is outlines in change 81.1.

4.1e) Rename Some Options

The HELP OPTION NAMES should be standardised, but be available through mnemonic names as well, for example:

SILENCE

and

REACT

should be accessable through the names

SILENCE

and

NO SILENCE (or REACT does exactly the same)

Along the same line:

BRIEF ---> BRIEF

VERBOSE ---> NO BRIEF (or VERBOSE or NORMAL all do exactly the same

This standardisation using NO as the negative of all options that are orinally set for the user saves the relisting of the options to find out the single proper key word to manipulate them. The mnemonic words (i.e. REACT and VERBOSE) gives simply another way to access the desired system action.

BEST COPY AVAILABLE

A second alternative system, incorporating the access structure mentioned in section 4.1b, would use the following names:

	VIEW (instead of SILENCE and REACT)
ENABLE	TRACE
DISABLE	SPELL
	BEGINNER (explained in next section)
	EXPERT (instead of BRIEF)

4.1f) Possibly add a "Very Helpful", verbose beginners mode

The change proposed here is for another possible mode for SOLO be introduced. Currently, SOLO has a default mode that can be called "Normal" mode. The user can use "BRIEF" mode if s/he desires shorter messages and hence saved output time. A third mode is proposed which would be called "Beginner " mode. This mode would be controlled by a new HELP OPTION:

BEGINNER (+) Very helpful version of SOLO with lengthy explanatory messages that are helpful to beginning SOLO students.

NOBEGINNER (-) Normal error messages (NORMAL does exactly the same)

Messages would be longer, more explanatory and include direction to help features and manual references. The main idea is to simply add a line or two to existing messages in order to clarify

BEST COPY AVAILABLE

their meaning.

This BEGINNER mode could be the default mode for users until a certain threshhold number of connect minutes or session has been reached. At the login time of the next session following the passage of this threshold value, the user could receive a special message alerting him/her to the other possible mode, i.e., NORMAL, and suggest that the user switch into that mode. Once that decision has been made by the user, NORMAL mode becomes the default login mode. (See general chage G.14.)

4.1g) Add a lesson in HELP usage

Although no actual frequency of occurrence was available for this analysis, the use of the extended HELP information appears to be quite minimal. Improved and a larger number of extended HELP messages, combined with an explicit lesson in the set of these messages would strengthen this feature of the SOLO environment as a user aid. The user is advised of the existence of these HELP messages, but never gets the hands on experience of using them. If this experience were made mandatory for beginning users, it is predicted that it would become a more widely used resouce during SOLO sessions. This prediction is based on the general principle that people do what they have done before, and are less likely to experiment with new items if confused on an existing item. The lesson should present some common errors, and the prompt to type HELP for more information. The user should ac-

BEST COPY AVAILABLE

tually type in HELP, and receive the extended message which the example would generate. An example of a specific procuedural HELP message, such as:

HELP DESCRIBE

should also be mandatory. This session should end with a final listing of the available extended help messages, and an explanation of the new normal format for prompting for those messages which would be:

[standard error message] (HELP?)

In BEGINNER MODE this message would be;

[standard error message]
[additional illuminating lines]
(Type HELP for more information concerning this error)

Two extended messages should be added to the HELP file:

HELP PARAMETERS
HELP *

These messages would deal with specific common conceptual problems surrounding these constructs, and offer solutions and primer page references.

The overall goal of all the above mentioned changes is to reinforce and improve existing SOLO features. These features pro-

BEST COPY AVAILABLE

vide other options for information gathering and trouble shooting of syntactic and conceptual problems. This helps to bring SOLO closer to being a truly friendly programming environment for the novice programmer.

BEST COPY AVAILABLE

4.2) Type Related Changes

The following pages cover each type of error which occurred in the data, and assess the ability of the current implementation to intelligently deal with the type. The format for the types is explained on page 23 of the methods section (section 3).

If a change is necessary, that change is specified and alternatives are presented.

NOTE: Types 15, 16, 45, 54, 78 and 79 have been discarded and hence unreferenced.

BEST COPY AVAILABLE

t1) Too many arguments to a procedure
e.g. NOTE FIDO LIKES WEST HAM UNITED
number of occurrences: 52
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low RTC: 2 EST: 8.40

Change 1.1: Intelligently handle 'too many arguments' to NOTE and FORJET.

Give an explicit message about the nature of triples and the fact that the user is only allowed to use triples in the database. This message could be added to the end of the existing message with a bracketed version of explicitly what the student typed in. (e.g. "You gave 'FIDO ISA BIG FAT SISSY' and that is too many nodes"). This retyping of the message is helpful for beginners in that it forces them to re-examine their input for errors, after SOLO has made its changes (i.e. dropping out illegal characters, although these changes should be mentioned to the user explicitly as well). A hypothetical example of this change at work follows:

```
SOLO: NOTE FI;DO ISA NASTY HEATHEN
 Oops... the format for NOTE is:
  NOTE node---relation--->node
 You gave:
  NOTE FI DO ISA NASTY HEATHEN
    (SOLO changed the ; to a space because it didn't recognise
    it as a viable character in this context).
```

This change would effect 29 occurrences.

Change 1.2: Intelligently handle 'too many arguments' to LIST, DESCRIBE & EDIT.

Give an explicit message about the SOLO procedures (i.e. LIST DESCRIBE & EDIT) which operate on single arguments at a time. This would improve the message to users who attempt to give multiple arguments to these procedures. A specific message could be added to the existing message:

BEST COPY AVAILABLE

```
SOLO: LIST PRAISE FOO IMPLICATE
 Oops: the format for LIST is:
  LIST procedure
 (LIST can only be used on one procedure at a time.  SOLO will
 LIST your first choice now, you will have to LIST FOO and
 LIST IMPLICATE separately).
```

This change would effect 16 occurrences.

t2) Spell corrector fault giving more than allowed number of arguments

e.g. NOTEW FIDO LIKES BEER
(The spelling corrector turns this into: NOTE W FIDO LIKES BEER and in turn produces an error message for the wrong format)

number of occurrences: 20
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low RTC: 2 EST: 3:20

Change 2.1: Link spell corrector to argument-counting demon

Change spelling corrector to access an argument-counting demon and determine whether to make a split of the extra character attached to the procedure name or to simply offer to drop the letter from the input line. If the split would cause a format error, then ask the user for confirmation before dropping the letter.

e.g.:

```
SOLO: NOTEW FIDO ISA DOG
      (when you typed NOTEW did you mean to type NOTE?):
```

This change would effect 20 occurrences of this error type.

BEST COPY AVAILABLE

t3) The user adds an unwanted space to the node/relation or procedure name after showing that s/he understands the format
e.g. NOTE FIDO LIKE S BEER
NOTE ONE IS A NUMBER
number of occurrences: 44
syntactic category: spelling error
status: trapable
priority: low RTC: 2 EST: 2 7:20

Change 3.1: Add 'pair-crunching' and argument-counting demon to spell corrector.

Change spelling corrector to work in special modes, trapping spelling errors of HELP OPTIONS input by user. Some of these errors (7) will be trapped by this change. The remaining 37 appear to be due to having the spelling option at "NOSPELL" setting, or using new node/relation names to which the corrector has no access. A clever use of the spell corrector in conjunction with an argument-counting demon, would be the following: first flag that there were too many arguments (see change 2.1), then scan the database for node and relation names in the triple which existed already, then suggest a condensation of the two unknown arguments:

SOLO: NOTE FIDO APPRECIATE S BEER
When you typed APPRECIATE S
did you mean APPRECIATES?

SOLO would not have known the relation name in this case, but would have sound the nodes in the database, and attempted the condensation or 'pair-crunching'. A simple trigger for the spelling corrector to become active, even in NOSPELL mode, would be a format flag from the argument-counting demon.

BEST COPY AVAILABLE

t4) Flow of control statement taken as an argument
e.g. if present: NOTE FIDO ISA DOG CONTINUE
number of occurrences: 42
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium RTC: 3 EST: 21

Change 4.1: Intelligent parsing of conditional lines

Intelligent parsing of lines that contain control flow statements. This would include the following features:

a) look for any possible match to a legal control flow word, regardless of spell corrector setting.

b) parse the line right to left to get this match so that:

Prini X continue

doesn't get taken as a print statement and reprompting for control word occurs.

c) have automatic correction of: to be ; making legal delimiters out of near miss delimiters.

d) remove any requirement for including the ; in the syntax, but have SOLO add it for the user, and show this change explicitly.

All occurrences of this error (42) could be pre-empted with these changes.

t5) Too few arguments given to a procedure
e.g. NOTE JANE LIKES
number of occurrences: 84
syntactic category: not enough args
conceptual category: none
status: improve message
priority: medium RTC: 2 EST: 14.00

Change 5.1: Add disambiguating message for conceptual error of 'too few arguments'

Possibly add a message specifying the necessity of proper number of arguments. These occurrences (89) all appear to be genuine errors not

BEST COPY AVAILABLE

based on misspellings but simply leaving an argument off.

An example of this type and an improved response is:

```
SOLO: FORGET MARY LIKES
 Oops... the format for FORGET is:
  FORGET node---relation--->node
 (you must always use triples with FORGET)
```

t6) Too few args because of missing space between node and relation name

```
 e.g. NOTE FIDO LIKESBEER
 number of occurrences: 61
      syntactic category: spelling error
      conceptual category: none
      status: trapable
      priority: medium  RTC: 2   EST: 10:10
```

Change 6.1: Add "node/relation-splitter" with argument counting demon

Use an argument-counting demon to trigger the spell corrector. Have the spell corrector look for the most likely break of the existing nodes and relations. If a reasonable split can be suggested, (based on previously-known node and relation names in the database), do so and ask for confirmation.

```
SOLO: NOTE FIDO LIKESBEER
 (when you typed LIKESBEER did you mean LIKES BEER?):
```

This change would trap all 61 occurrences of this type of error.

t7) No arguments given

```
 e.g. NOTE
 number of occurrences: 67
      syntactic category: args problem
      conceptual category: none
      status: ok
      priority: na
```

Change: NONE

Error type occurred 67 times. Possibly add the "Type Help" for more information message in the 'BEGINNER mode'. (Described in changes to HELP OPTIONS, section VI).

t8) Errors due to type ahead problems
e.g. NOTE FIDO ISA DOG FORGET FIDO LIKES BEER
number of occurrences: 99
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3 EST: 49:30

Change 8.1: Add input cut-off to reduce type ahead errors Occurred 99 times in different contexts. This problem would decrease as reaction time of the implementation and system decreased, or with the re-implementation of SOLO on micros. One improvement which would be possible is the introduction of a cut off. Once the user had made a type ahead error in a forced choice situation (e.g. confirmation requested by node-name spelling corrector, control flow, prompter, or HELP OPTION selection), the following input lines would be thrown away. Only input received after part of the repeated line had been presented would be accepted. This CHANGE would save multiple type ahead errors in a row, but might interfere with the user model of how SOLO deals with type ahead. Yet at 30 characters per second, this could save substantial wait and connect time for recovery from a series of type ahead errors.

BEST COPY AVAILABLE

t9) Used "?" in a node in a NOTE or FORGET
e.g. NOTE FIDO ISA ?
number of occurrences: 19
syntactic category: formal/actual variable name reversal
conceptual category: variable name confusion
status: improved message
priority: medium RTC: 3 EST: 9:30

Change 9.1: Trap special format error and give clarifying message

Improve message to point out the fact that this case of a format error is special. "?" has no meaning here and perhaps, if they are in edit mode, they wished instead to type *. An example of an extended 'BEGINNER mode' message follows:

```
SOLO: NOTE FIDO ISA ?
 Oops:... the format for NOTE is:
  NOTE node---relation--->node
 (The "?" you used has no meaning here.  It is only used in
 the CHECK statement).  Type HELP NOTE or HELP CHECK for more
 information)

                        or

...: 3 FORGET FIDO ISA ?
(when you typed '?' did you mean '*' (yes or no): No
 Oops... the format for FORGET is:
    FORGET node---relation--->node
(the "?" has no meaning used here.  I is only used in the
CHECK statement.  Perhaps you should have used "*" instead
of "?").
```

The second example above could be made more intelligent by making the final sentence of it's error message contingent on the use of the "?" in the procedure before the line on which this error occurred. Only in this case would * be possible later in the procedure, and hence a possible confusion of ? and *. (see change 48.1)

t10) Illegal characters in node or relation names
e.g. FORGET FI;DO LIKES BEER
number of occurrences: 7
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

t12) Spurious control characters in node or relation names
e.g. FOR EACH CASE OF FIDO IS^A DOG
number of occurrences: 36
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high
(two classes of the same error type which have been combined for this analysis)
total number of occurrences: 43 RTC: 2 EST: 7:10

Change 10.1: Intelligently handle illegal characters using argument counting demon

Instead of reading illegal characters as spaces, delete them from the line. The decision could also be contingent upon the argument-counting demon. If the arguments come out to be the wrong format, then interpret the illegal character as a space or non-space as is appropriate:

SOLO: NOTE FI^DO ISA DOG
(illegal character "^"; line read as:
NOTE FIDO ISA DOG)

or

SOLO: NOTE FIDO^ISA DOG
(illegal character "^", line read as:
NOTE FIDO ISA DOG)

The message might be better phrased as "you typed the character "X" which SOLO can't understand in this statement, so it read the line as..." or this message could be the extended HELP message for this error.

This change would trap all 43 occurrences of these errors.

BEST COPY AVAILABLE

t11) Over legal limit for number of characters in a node or procedure name
e.g. NOTE RICHARD ISA WISHYWASHYONE
number of occurrences: 36
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high RTC: 4 EST: 3:00:00

Change 11.1: Add special message about name length limit or extend limit

Change either:

a) The limit to be greater than 12. 24 has been suggested, and no occurrences of node or procedure names which exceed that limit have been encountered. If 24 characters are used, allow '.' or underline ('_') to be a legal character, e.g. 'To WORK_ON /X/'.

b) The message. The current message is a format error message which is completely misleading. If a special message were displayed which stated the true cause of the error and warned the user, then all occurrences (36) would be intelligently handled.

e.g.

SOLO: TO COUNTBACKWARDS /FIRSTNUMBER/
Oops... the procedure name you chose is too long for SOLO to handle. Try using another name or a contraction of the name you tried, but make sure that it is 12 characters or less in length.

This could also be adapted to be an extended help message after the student is out of the beginners mode.

t13) Steamroller effect by the synonym and spelling corrector
... Compare /X/ /Y/
(COMPARE=CHECK)
Oops... the format for CHECK is:
number of occurrences: 26
syntactic category: none
conceptual category: none
status: trapable
priority: high RTC: 5 EST: 6:30:00

BEST COPY AVAILABLE

Change 13.1: Turn off synonym feature of spell corrector.

Turn off the synonym changing feature of the spelling corrector. It worked successfully only two times, missed 4 possible cases where it could have been useful and "steam rolled", or made an unwanted, counterproductive change in 26 cases. Since synonyms can cover a wide range for each SOLO procedure name and this range intersectswith possible user defined procedure names, it is suggested that no attempt to look for synonyms be made. If the user uses a synonym, s/he will be presented with an error message and be made aware s/he is using a construct which doesn't exist. It is assumed that then the user will reference the rear cover of the primer for the correct construct desired.

This ineffectiveness of the synonym finder is combined with the disorienting effect of having an unwanted change made for you (usually accompanied by an error message, when the user's input line was, in actuality, flawless) to make a very strong case for the discontinuation of the synonym feature of the spelling corrector. (Note: this change has recently been implemented).

t14) Over extended use of wild card
e.g. FOR EACH CASE OF ?---LIKES--->BEER
number of occurrences: 7
syntactic category: variable constraint violation
conceptual category: generalizing variable positioning validity
status: improved message
priority: medium RTC: 4 EST: 35:00

Change 14.1: Trap and clarify concept behind over extended use of wild card.

An improved or extended error message should be included to handle this special case of a CHECK format error. The message should include a reference to the "one-way-ness" of inferencing in SOLO and a page reference in the primer to clarify the error, e.g.

BEST COPY AVAILABLE

```
SOLO: CHECK ?  ISA PERSON
 Oops... the format for CHECK is:
   CHECK node---relation--->?

                or

        CHECK node---relation--->node
```

(You used the ? in the first position instead of the required last position of the node/relation triple. This is a clever idea, but unfortunately not allowed in SOLO. See pages 16-18 for information about "one-way" inferencing). This would improve the message for all 7 occurrences of this error.

t17) Wrong access to the help features
e.g. HELP CAT
number of occurrences: 8
syntactic category: wrong mode for input
conceptual category: help mode accessible directly from SOLO top level
status: trapable
priority: medium RTC:3 EST: 4:00

Changes are discussed in the special section on HELPOPTIONS, presented previously.

t18a) Mode confusion; student assumed s/he was in top level
e.g. ...: LIST
number of occurrences: 42
syntactic category: none
conceptual category: mode error
status: improve message
priority: medium RTC: 3 EST:

Change 18a.1: Special message for EDIT mode error

If the user inputs a built-in SOLO procedure which is an obvious mode error (i.e. LIST or TO) the system could have a mode-demon to identify this as a special error and present a message about the

BEST COPY AVAILABLE

nature of the error and how to recover from it, e.g.:

```
...: LIST PRAISE
 Oops... you can't LIST a procedure while you are editing or
 defining a procedure (type HELP for more information)
...: HELP
```

If you would like to LIST PRAISE, you will first have to finish editing the procedure which you are currently editing, by typing: DONE. Then use the LIST PRAISE command after the 'SOLO': prompt. Until you have typed DONE you will remain editing this procedure.

```
...:
```

This proposed change would improve the message for all 42 occurrences of this type of error.

Change 18a.2: More flexible LIST

A second alternative is to allow students to list their procedure while in EDIT or DEFINE mode. This proposed change is well documented in a set of supplementary SOLO additions, for 1980 Summer School users by Marc Eisenstadt. The flexible list is defined on page 23 of this document, which is included as appendix C in this paper.

t18b) Mode confusion: student assumed s/he was in editor
e.g. SOLO: 2 FOR EACH CASE OF X Y ?
number of occurrences: 21
syntactic category: none
conceptual category: mode error
status: improve message
priority: medium RTC: 3 EST: 10:30

Change 18b.1: Special message for top level mode error

Change error handling. It was clear from the surrounding context in most cases that the user thought s/he was in the editor and wished to add the line to a procedure. Instead of merely throwing the line number away and performing the statement, it is suggested

BEST COPY AVAILABLE

that the error message states the possible intent of the user and doesn't effect the database, e.g.

```
SOLO: 3 NOTE FLAG---IS--->SET
 Oops... you included a line number on that line.  If you
 want to add it to a procedure, then you must create or edit
 that procedure then type the line.  If you wanted to simply:

    NOTE FLAG IS SET

  then retype that command, but leave out the line number.
```

This assumes the constraint of having procedure names not legally begin with numbers. This change would improve the message of 21 occurrences of this error type.

A shorter version of this message would be: "Oops... you have a line number on this line, but you're not defining or editing a procedure."

t19) Attempt to apply a procedure to a procedure
e.g. FORGET TO FOO or FORGET FOO
number of occurrences: 1
syntactic category: violation of defined syntax
conceptual category: need to 'obliterate' unwanted procedure
status: improved message
priority: medium RTC: 2 EST: 00:10

Change 19.1: Explain that FORGET cannot work on procedures.

Improve message by trapping this attempt to erase a procedure. The user wanted to erase or re-define the procedure and those options would be suggested in a specially tailored message to meet this case:

```
SOLO: FORGET TO FOO
 Oops... you can't FORGET a procedure like FOO.  If you would
  like to rewrite it, or get rid of it completely just
  type EDIT FOO then wait and make the desired changes.  Delete
  all lines to delete the procedure
```

BEST COPY AVAILABLE

Although there was only one occurrence of this type of error in our dribble files, we have actually witnessed several occurrences of this and had about 15 students ask 'How can I get rid of a rubbishy definition', so there seems to be a genuine demand for some sort of 'obliterate' capability (see change 19.2).

Change 19.2: Incorporate an 'obliterate' primitive.
Have a procedure eraser called obliterate,
e.g.

```
SOLO: OBLITERATE FOO
 OK... I have obliterated the definition of the procedure FOO.
```

t20) Attempted concatenation of strings
e.g. CHECK X "A"/ANY/ ?
number of occurrences: 4
syntactic category: putting strings into node spaces
conceptual category: thinks that concatenation is possible
status: trapable
priority: low RTC:5 EST: 01:00:00

Change 20.1: Trap and explain illegality of attempted concatenation. This change would require a specialized demon to look for quoted strings in node/relation triples. If one was found, and the argument counting demon had found a format error, then the quoted string demon could send off a message explaining the nature of the conceptual error the user was exhibiting. Since this error occurred only 4 times, between only two users the priority is quite low to have this change implemented. An example of this specially trapped error message would be as follows:

```
...: 4 CHECK /X/---ISA--->"Big"*A
Oops... you aren't allowed to make new node names like "Big"*A
by combining node names and quoted strings.
```

BEST COPY AVAILABLE

t21) Use more than three args to a user defined procedure
e.g. TO LOOKUP /W/ /X/ /Y/ /Z/
number of occurrences: 14
syntactic category: too many args to a procedure
conceptual category: thinks that an unlimited number of args are possible
status: ok
priority: na

Change: NONE

This message handled all 14 occurrences of this error type successfully and the users appeared to comprehend their mistake quickly (on the whole) and change to using 3 or less parameters. An alternative would be to allow an unlimited number of arguments to a procedure.

t22) Forgot "/" in the procedure line during definition
e.g. TO LOOKUP X/
number of occurrences: 3
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: high RTC: 2 EST: 00:30

Change 22.1: Add missing slash to parameter name in title line.

Simply add the opening "/" for the student, but state this explicitly and retype the line in beginners mode, e.g.

```
SOLO: TO PRAISE X/
 Oops... you forgot the first /
 I'll add it and we'll continue
TO PRAISE /X/
 ...:
```

This change though seeming trivial, could be quite useful to the beginning student. This change pre-empts all 3 occcurrences, and seems to be an easy implementation. A general addition to this message in 'BEGINNER mode' would be to add a reminder line that the parameter must be surrounded by slashes, e.g.

BEST COPY AVAILABLE

BEST COPY AVAILABLE

```
SOLO: TO FIND firstperson secondperson
 Oops... the format for TO is:
  TO procedure (/parameter1/ /parameter2/ /parameter3/)
  (don't forget to put /'s around the parameter names)
```

t23a) Gave procedure name with TO attached to it
e.g. LIST TO FOO
number of occurrences: 14
syntactic category: included procedure definition term in procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high

t23b) Gave procedure name with args included to a procedure
e.g. LIST FOO /X/ /Y/
number of occurrences: 9
syntactic category: included args as part of procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high
total number of occurrences: 25 RTC:3 EST: 12:30

Change 23.1: SOLO procedures should accept TO or parameters with procedure names.

Change the SOLO procedures to accept this added information, but with a warning about the non-necessity of the TO and parameters.

```
SOLO: LIST TO FOO
      (the TO that you added to your procedure name FOO above
      is not necessary here.  You need only refer to the procedure
      by its name:

      LIST FOO
OK...

          or

SOLO: LIST FOO /X/ /Y/
      (the parameters /X/ and /Y/ that you added to the
      procedure name are not necessary here.  You need only refer
      to the procedure by its name:

      LIST FOO
OK...
```

These are relatively common, 'over-cautious' beginners errors and should be safely trapped. This change would pre-empt all 25 occurrences of this type of error.

t24) Editing node names
e.g. EDIT FIDO
number of occurrences: 3
syntactic category: giving date as an argument in place of a procedure
conceptual category: data/procedure confusion, argument requirement confusion
status: improve message
priority: medium RTC: 3 EST: 01:30

Change 24.1: Trap and explain 'EDIT node name' error.

Improve message for BEGINNER by giving more information about the possible conceptual problem behind this error, e.g.

```
SOLO: EDIT FIDO
Oops... you can't EDIT a node name like FIDO.
If you'd like to change the description of FIDO, use NOTE
and FORGET
```

This would improve all three occurrences of the error, but it isn't a high priority change.

BEST COPY AVAILABLE

t25) Forgot " which SOLO then correctly added
e.g. PRINT "FIDO MUST BE A DOG
number of occurrences: 115
syntactic category: missing delimiter
conceptual category: none
status: ok
priority: na

Change: NONE

SOLO correctly helped delimit print strings 115 times. However, the change should be made, then printed so the user can see how SOLO has helped correct their code;

SOLO: PRINT "JEFF IS GREAT
(Right hand quotes (") assumed and added)
PRINT "JEFF IS GREAT"
JEFF IS GREAT

t26) Forgot to include an obvious SOLO procedure name
e.g. SOLO: 6---ismorethan--->7
number of occurrences: 60
syntactic category: missing procedure name, arguments only
conceptual category: none
status: trapable
priority: medium RTC: 2 EST: 10:00

Change 26.1: Prompt for or give obvious, forgotten SOLO function. In the case of an obvious CHECK format, the change could be made for the user without confirmation, e.g.

SOLO: FIDO---ISA--->?
(You forgot to include the procedure name: CHECK)
CHECK FIDO---ISA--->?
Present... DOG

In the case of just a recognizable triple being input at top level, any of NOTE FORGET or CHECK could be the possible intention of the user. One possible way to handle this situation would be to present several choices:

BEST COPY AVAILABLE

```
SOLO: FIDO---ISA--->DOG
 Oops... you forgot the procedure name, did you want to:
 NOTE FIDO ISA DOG into the data base? (yes or no):
```

and continue checking the other two possibilities in the same manner. A look at the data base could guide the order of presentation of these alternatives. If the triple already existed in the database, ask first if s/he wanted to FORGET it, then if s/he wanted to CHECK for its existence, and finally if s/he wanted to NOTE it. The reverse is true if it doesn't exist in the database. This change would pre-empt all 60 occurrences of this error type.

t27) Gave single number at top level
e.g. SOLO: 6
number of occurrences: 13
syntactic category: no procedure name given or defined for the number
conceptual category: perhaps a hanging input mode confusion
status: improve message
priority: high RTC: 2 EST: 02:10

Change 27.1: Trap and explain single number error.

This is a serious, high priority error to trap. Currently the input of a single number at top level gets a BASIC error message back to top level SOLO. With the constraint that procedures cannot begin with numbers (see general change number G1.1), this leaves only a few alternative explanations for this error.

This first possibility is a simple typing error. This error could be clarified with an improved message that would also clarify the other possible source of this error; that of a mode error. The user might believe that s/he is in edit mode, removing lines by typing single numbers. A message such as the following would handle both cases of this error type:

BEST COPY AVAILABLE

SOLO: 6
Oops... a single number like 6 doesn't mean anything to SOLO, unless you are defining or editing a procedure (see page 43 of the primer for more information or type HELP EDIT)

This change covers all 13 of the occurrences of this error type.

t28a) No referent
e.g. DESCRIBE BASEBALL or LIST FOO
(there is no node called BASEBALL and no procedure called FOO)
number of occurrences: 201 (106 node names; 95 proc. names)
syntactic category: none
conceptual category: possibly a missing desired effect or unexpected side effect, or could be deliberate
status: ok
priority: na

t28b) Describe a relation name with no referent
e.g. DESCRIBE LIKES
(there is no description of LIKES in the data base)
number of occurrences: 29
syntactic category: none
conceptual category: possible node/relation confusion; often done deliberately
status: ok
priority: na
total number of occurrences: 230

Changes: NONE

The people seem to be attempting to access information which doesn't exist. The current message appears to be sufficient to handle all 230 occurrences of the error. This is an example of a simple error which is demonstrative of a higher level error, such as being confused as to why certain triples are not being added to the data base by the procedure designed to do so.

BEST COPY AVAILABLE

t29a) Attempt to use DESCRIBE on a procedure
e.g. DESCRIBE FOO
number of occurrences: 24
syntactic category: mixed arg types
conceptual category: data/procedure confusion, concept of a general "show me" procedure
status: improve message
priority: medium

t29b) Attempt to use LIST on a node
e.g. LIST FIDO
number of occurrences: 11
syntactic category: mixed arg types
conceptual category: data/procedure confusion, concept of a general "show me" procedure
status: improve message
priority: medium
total number of occurrences: 25 RTC: 3 EST: 12:30

Change 29.1: Trap and explain data/procedure confusion.

Have a demon which looks for the match of argument type with format type. If the person attempts to give data in a procedure format slot, explain this error specifically. If the reverse is true, inform the user as well, e.g.

SOLO: DESCRIBE PRAISE
Oops... you can't use DESCRIBE on a procedure like PRAISE. If you'd like to look at PRAISE use LIST PRAISE (type HELP LIST for more info) To activate PRAISE, type only the name after the SOLO prompt.

or

SOLO: LIST FIDO
Oops... you can't LIST a node like FIDO. If you'd like to look at the node FIDO and its relations use DESCRIBE FIDO (type HELP DESCRIBE for more info)

These improved messages would cover the 35 occurrences of these error types, and hopefully reinforce the concepts of data, procedures, and the distinction between the two.

BEST COPY AVAILABLE

t30) Use Describe as an invocation function
e.g. DESCRIBE DEMO1
number of occurrences: 9
syntactic category: procedure arg in place of data arg
conceptual category: necessary invocation program
status: ok
priority: na RTC: 3 EST: 04:30

Change 30.1: Trap and explain data/procedure confusion.

Same as 29.1 above. The final line of the first example was added to trap this error type. If implemented, the change would successfully cover all of the 9 occurrences.

t31) Attribution of too much knowledge to a function
e.g. DESCRIBE SOLO or DESCRIBE NUMBER
number of occurrences: 2
syntactic category: improper arg type
conceptual category: user thinks system has human understanding capabilities
status: ok
priority: na

Change: NONE

This error type is also best handled by change 29.1. Since the possibilities for this type of error are limitless, it would be impossible to trap them consistently, hence no attempt should be made to second guess the users' intentions in this case.

BEST COPY AVAILABLE

t32) Spelling error
e.g. HELP OPTION
(SOLO requires HELP OPTIONS)
number of occurrences: 93
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2 EST: 15:30

Change 32.1: Turn on spelling corrector in fancy features.

Change the spelling corrector to operate while in special features mode (i.e. help options, confirmation situations , control flow, and statement addition mode). This would trap 33 of the total 93 spelling errors. Having a demon to search for legal placement of : (instead of ;) in an if present/if absent branch of a statement and making that substitution would trap another 11 occurrences of this error type. The remaining 49 spelling errors are attributable to first occurrences of the node or relation name, and the fact that the spelling corrector was turned off in some cases.

t33) Too many lines in procedure
e.g. (student uses more than 10 numbered lines)
number of occurrences: 12
syntactic category: none
conceptual category: overflowing a present limit
status: ok
priority: na

Changes: NONE

The message seems to adequately handle the error type, and users responded appropriately to the message by removing lines, or writing sub-procedures. There were 12 occurrences of this error type.

A suggested addition would be an extended HELP message regarding this limit and its pedagogical effects and rationale.

BEST COPY AVAILABLE

t34) "*" /"?" confusion
e.g. FOR EACH CASE OF FIDO ISA *
number of occurrences: 13
syntactic category: variable/value reversal
conceptual category: variable/value concept or name confusion
status: trapable
priority: medium RTC: 3 EST: 6:30

Change 34.1: Trap and disambiguate "*"/"?" confusion in lines.

This change is the same type as specified in change 9.1 and would trap all 13 occurrences in this case.

```
...: FOR EACH FIDO ISA *
when you typed "*" did you mean "?" (yes or no): No
 Oops the format for FOR EACH is:
  FOR EACH node---relation--->?
(* has no meaning used here.  Perhaps you meant to type "?"
instead of "*".  For more information type: HELP FOR EACH
or refer to page 84 of the primer)
```

Alternatively, the change could be made for the user, and the line reprinted with a warning i.e.

```
... FOR EACH FIDO ISA *
(you can only use ? in this statement)
...: FOR EACH FIDO ISA ?
.....
```

This question of which alternative strategy to use here centers around the issue of which alternative reinforces the concept most strongly, and also is most time efficient. The suggested alternative is the first alternative, at least for the BEGINNER mode.

BEST COPY AVAILABLE

t35) FOR EACH CASE OF using only a triple
e.g. FOR EACH CASE OF FIDO ISA DOG
number of occurrences: 32
syntactic category: using a constant for a variable slot
conceptual category: iteration control error
status: improved message
priority: medium RTC: 3 EST: 16:00

Change 35.1: Trap special case of FOR EACH error and explain. Improve message by adding a description of the error and a help reference for this specific error, i.e.

```
...: 3 FOR EACH FIDO ISA DOG
 Oops... the format for FOR EACH is
     FOR EACH CASE OF node---relation--->?
(you gave the node dog in the third position instead of
 a "?".  Type HELP FOR EACH for more information)
```

There were 32 occurrences of this error. This relatively high number is due to the use of FOR EACH like a CHECK statement. CHECK can be used either for constant or variable values, while FOR EACH must be used with variable value, even though a user might be sure that there is only one "FIDO---ISA" relation in the database. An alternative is to legalise the construct so that it can be used like a CHECK statement. This remains consistent with the syntax and concept behind the wording of the statement. There will only be one case of FIDO ISA DOG, and the operation could be carried out only if that specific data construct is in the data base.

t36) Attempt to get out of fancy features
e.g. type option here: DONE or END
number of occurrences: 2
syntactic category: arg not in specified arg set
conceptual category: escape form mode error
status: trapable
priority: medium RTC: 2 EST:00:20

Change 36.1: Restructure HELP OPTIONS

These errors (2) will be pre-empted by the restructuring of the

BEST COPY AVAILABLE

HELP OPTIONS access, (see changes to HELP OPTIONS, section 4.1)

t37) Attempt to use COPY as a HELP option
e.g. type option here: COPY
number of occurrences: 2
syntactic category: wrong input error
conceptual category: mode/feature access error
status: improved message
priority: low RTC: 3 EST: 01:00

Change 37.1: Trap and explain 'wrong access' error.

Improve the message and add a demon to the HELP OPTIONS which looks for this error and directs the user back to the top level of SOLO to carry out the desired COPY command:

Type option here: COPY
Oops... COPY is not a HELP OPTION
Use the COPY command like other built in SOLO procedures
(like NOTE, FORGET etc.)

2 occurrences.

t38) Student in infinite Loop
e.g. (student simply defines a recursive procedure with improper flow of control)
number of occurrences: 64
syntactic category: none
conceptual category: flow of control error
status: ok
priority: na

Changes: NONE

Existing message is clear and refers the user to the extended HELP information option. The existing message handles all 64 occurrences of the error, though users exhibit a behaviour in this situation

BEST COPY AVAILABLE

which is similar to a "If I close my eyes, maybe it will go away" behaviour and usually run the looping program at least once more on the same example to see if it "really does" get into an infinite loop.

t39) Attempt to COPY a FOR EACH procedure before state change to FOR EACH mode
e.g. COPY FOO C#123812 (FOO has a FOR EACH in it)
number of occurrences: 2
syntactic category: none
conceptual category: possibly doesn't understand the change in state of the system form wildcard to FOR EACH state
status: ok
priority: na

Changes: NONE

The message covers the error explicitly. An extended help message might give references to the relevant primer pages that cover the FOR EACH mode change. Only two occurrences of this error type appeared.

t40) Mixed syntax in subline prompt
e.g. .... 10b If Present:
(should be "10a IF Present:" or "10b If Absent")
number of occurrences: 1
syntactic category: mixed syntax
conceptual category: possibly inferring too much flexibility to the syntax
status: ok
priority: na

BEST COPY AVAILABLE

Changes: NONE

The only occurrence (1) of this error type seemed to be handled by this message, however, a full explanation should be added to the extended help file.

"You have mixed up the line letter (i.e. A or B) and the syntax for that line. For example another student might have typed:

... 3A If Absent
and SOLO requires the "A" line (first line of the conditional) to be "If Present". Check for a similar mistake in your line.

t41) Use of wild card in check, while in FOR EACH mode
e.g. SOLO: CHECK FIDO ISA ?
number of occurrences: 21
syntactic category: none
conceptual category: not understanding the change in state of system
status: ok
priority: na

Changes: NONE

This message and extended HELP message seem to effectively handle all 21 occurrences of this error type.

t42) User record not found in the data base
e.g. (title is self explanatory)
number of occurrences: 3
syntactic category: possibly a mis-typed student
conceptual category: none
status: ok
priority: na

Change: NONE

Error message applies to the 3 cases of this error type. In addition, this feature is generally available only to priviledged

BEST COPY AVAILABLE

users, not to students.

t43) Non-unique variable names in a global variable situation
e.g. (person uses a "*" variable in more than one FOR EACH statement)
number of occurrences: 4
syntactic category: non-unique variable name referenced
conceptual category: locality of variable problem possibly
status: ok
priority: na

Change: NONE

Possibly add an extended help message, but this message appears to have been explicit enough to point out the error in all 4 cases where it occurred. Refer to page 90 of primer if extended help is called for.

t44) FOR EACH or * used at top level
e.g. SOLO: FOR EACH CASE OF X---Y--->?
number of occurrences: 2
syntactic category: wrong mode for the construct
conceptual category: mode confusion, or misunderstanding constraints of construct
status: ok
priority: na

Change: NONE

Perhaps add extended help message explaining why a subline prompt (e.g. 10A) can not be issued at top level.

BEST COPY AVAILABLE

t46) Use of control flow statements where inappropriate
e.g. 1 PRINT "BLAH"; CONTINUE
(CONTINUE is inappropriate here, and SOLO attempts to put it into the PRINT statement)
number of occurrences: 82
syntactic category: inappropriate use of control flow statement
conceptual category: control flow specification possible or necessary at any point in the procedure
status: trapable
priority: medium RTC: 4 EST: 06:50:00

Change 46.1: Trap and explain illegal control flow statements. Have a control flow word demon that is triggered by a format error or works automatically for PRINT statements. This is quite a common error (84 occurrences) and points up a serious conceptual error about where control statements appear and what they do. A special message should point out that the control statement is not needed and then give the student the line if the format is correct:

```
...: 3 NOTE FIDO---ISA--->DOG ; CONTINUE
 Oops... the "; CONTINUE" was not necessary. Use of EXIT,
 CONTINUE and NEXT CASE is only necessary in CHECK and FOR EACH
 constructs.  I'll remove it for you, and retype the line so
 that you can continue:
...: 3 NOTE FIDO---ISA--->DOG
...:
```

or

```
...:5 PRINT "END OF PROCEDURE"; EXIT
 Oops... the "; EXIT" was not necessary here.  Use of
 EXIT, CONTINUE and NEXT CASE is only necessary in CHECK
 and FOR EACH constructs.  I'll remove it for you and retype
 the line so you can continue:
...: 5 PRINT "END OF PROCEDURE"
...:
```

Previously, in the first case there would have been a format error. In the second SOLO would have attempted to put the EXIT in quotes and include it in the print statement quoted string.

BEST COPY AVAILABLE

t47) Delimiter problem
e.g. If present: PRINT "OKAY";/X/;CONTINUE
number of occurrences: 22
syntactic category: wrong delimiter used
conceptual category: none
status: improved message
priority: medium RTC:3 EST:11:00

Change 47.1: Trap and correct delimiter problems

People use the ";" delimiter as a general delimiter to break up parts of a print statement,(possibly an intrusion from BASIC). This throws off the current algorithm which looks for the control flow statement after the ";". As with other problems with control flow statements being taken as arguments, or generally not being read as control flow statements, the solution is to parse the input line from right to left looking for valid statements. Once one has been found, the remaining illegal ";"s can be thrown out. Quotes can be added where necessary (as outlined in change 25b) and the revised line reprinted for student evaluation, e.g.

```
.... 2A If Present: PRINT "OKAY"; /X/; "IS THE ONE"; CONTINUE
     (you had some illegal or unnecessary characters in that
     line.  I'll remove them and reprint the line for you)
.... 2A If Present: PRINT "OKAY" /X/ "IS THE ONE"; CONTINUE
```

or

```
.... 2A If Present: PRINT FINALLY OVER WITH /X/ CONTINUE
     (you forgot the ; I'll add it)
     (left quotes (") assumed and added)
     (right quotes (") assumed and added)
     (I'll reprint the line for you)
.... 2A If Present: PRINT "FINALLY OVER WITH" /X/; CONTINUE
```

t48) Confusion between /X/, X and *X as values in a PRINT statement
e.g. PRINT H X (should have been /H/ /X/)
(SOLO also throws away "/"s without mention if there is no parameter referent)
number of occurrences: 36
syntactic category: variable name/value/parameter denotation problem

BEST COPY AVAILABLE

conceptual category: variable name/value/parameter distinction
status: improved message
priority: high RTC: 4 EST: 03:00:00

Change 48.1: Trap and disambiguate parameter, variable and value problems in PRINT.

This is a common conceptual problem, particularly with beginners. Intelligent parsing is needed, with some specialised demons.

Demon 1: looks for parameters in title line (e.g. 'TO FOO /X/') matches to parameters in the body of the procedure.

Demon 2: Detailed in change 73.1, looks for previous calling of a variable by using the ? It also looks for the value of the variable without the * prefix (i.e. A instead of *A) and specifies any variable inconsistencies as they arise during the edit.

When a line containing PRINT is input, it is scanned by demons 1 and 2 for illegal variables and parameters calls. These are pointed out, with references to SOLO primer pages and extended Help information regarding some restriction and arguments passing up to parent procedures. An example where parameters are used in the body but not in the definition, follows:

```
SOLO: TO FOO
 ...: 1 PRINT "MY TAKE IS" /X/
 Oops... /X/ doesn't refer back to any parameter that you
 gave to the procedure FOO.  See pages 39-42 of the primer
 or finish editing by typing DONE and type HELP PARAMETERS

...:
```

or

```
SOLO: TO FOO
 ...: 1 CHECK FIDO LIKES ?A
 ...: 1A IF PRESENT: PRINT "A"

 (did you want to print the value of A (namely *A) that you
 found in the CHECK FIDO LIKES ?A) (yes or no): Yes

 ...: 1A IF PRESENT: PRINT *A
```

These two demons could effectively deal with the 36 occurrences of these errors and point the student to material which would help clarify their understanding of the print format and the difference

BEST COPY AVAILABLE

between variables and values,

t49) Multiple procedure calls on a single line
e.g. .... if present: PRINT "GOT IT": FOO: EXIT
number of occurrences: 14
syntactic category: illegal format
conceptual category: confusion about procedure call constraints
status: improved message
priority: medium RTC: 4 EST: 01:10:00

Change 49.1: Trap and explain illegality of multiple procedure calls on a line

Using a combination of right to left parsing, a demon which recognizes procedure names and a format demon, this specific error can be trapped and pre-empted with a warning that explicitly deals with the conceptual confusion which produced the error. After the line has been parsed and had illegal characters removed, quotes added and control statement present, (see changes 46.1, 47.1, 48.1) if a format error exists (ie. too many args to a NOTE) then the final arg could be checked to see if it is an already existent procedure name. If so, the error is one of this type and an appropriate change can be made and message presented that might solve the problem, e.g. the person wants to print a message, then run a procedure called FOO:

```
...: 3A IF PRESENT: PRINT "NOW TO DO IT"; FOO /X/; CONTINUE
     (Some illegal characters used here
     Oops FOO is a procedure, you're only allowed to have
     one procedure on a line.  You will have to write a separate
     subprocedure to do all the things which you desired to do
     on this line)
...3A If Present:
```

This would cover all the 14 cases of this error type.

BEST COPY AVAILABLE

t50) Positive response to a confirmation statement
e.g. Do you want to select any more options? YES
number of occurrences: 129
syntactic category: none
conceptual category: none
status: ok
priority: na

t51) Negative response to a confirmation statement
e.g. Do you want to select any more options?: NO
number of occurrences: 388
syntactic category: none
conceptual category: none
status: ok
priority: na
total number of occurrences 517 RTC: 2 EST:01:26:10

Change: None

This type would be excluded in the suggested revised implementation of the HELP OPTIONS. This suggested revision is strengthened by the fact that: no. of 'multi-option' uses of this mode: 129

no. of 'single-option' uses of this mode: 388
-129
=259

This proportion of single option uses (twice as many) suggests that the structure is faulty, and a single option use default structure should be established. This is detailed in the HELP OPTIONS revision.

BEST COPY AVAILABLE

t52) *A/A confusion in parameter list
e.g. TO FOO /*A/
Oops... Illegal characters [*] in parameter name ignored (*A=A)
(SOLO corrects this to to TO FOO /A/)
number of occurrences: 2
syntactic category: parameter syntax violation
conceptual category: variable/parameter confusion
status: improve message
priority: na RTC: 2 EST: 00:20

Change 52.1: Expand message to clarify *A/A confusion in parameter list

This message traps the error, but could be improved with an extended help message that deals with the conceptual problem which the person has. They appear to be passing values into a procedure instead parameters. They could be directed to the proposed HELP PARAMETERS extended help message for more information. This error occurred only 2 times and was trapped both times and corrected.

BEST COPY AVAILABLE

t53) Wrong procedure call to a student defined procedure.
e.g. FOO /ME/ /YOU/
(FOO only takes 1 arg in this case)
number of occurrences: 97
syntactic category: wrong number of args to procedure
conceptual category: none
status: improve message
priority: high RTC: 2 EST: 16:10

Change 53.1: Add student defined parameter names to message no. 14.

Improve message by adding the parameter names instead of referring to them as parameter 1, parameter 2, etc. This error occurred 97 times, and the existing message appears to be appropriate, specifying the number of parameters, but the actual parameter names would be more informative, e.g.

SOLO: PRAISE
Oops... the format for PRAISE is:
PRAISE /PERSON/
SOLO:

t54) Student leaves machine and fails to respond to two previous warnings about auto log out.
e.g. (8 minutes have elapsed. Please press the RETURN key).
(16 minutes have elapsed! Are you still there? Please press return key) (sorry... you will be logged off if you don't type something at once)
number of occurrences: 15
syntactic category: none
conceptual category: none
status: ok
priority: na

Changes: NONE

The existing message appears clear and succint. This message occurred 15 times.

BEST COPY AVAILABLE

t55) Attempted access to "built-in" SOLO procedure
e.g. LIST HELP
number of occurrences: 1
syntactic category: none
conceptual category: system procedure changes allowed
status: Trap
priortiy: na RTC: 3 EST: 00:30

Change 55.1: Trap specific case of listing help options and pre-empt.

The user could have been attempting to look at the code of the help file, and in that case, the present message is adequate, however there is another possibility. That is that the user wanted to "LIST" the available HELP OPTIONS or extended Help messages. A special trap could be implemented for this special case (which occurred only once), which would trap the occurrence:

SOLO: LIST HELP
Oops... you can't LIST built-in SOLO procedures (like NOTE, etc.)
If you wish to see the HELP OPTIONS or HELP INDEX, type this in after the "SOLO" prompt.

t56) Attempt to EDIT a built in SOLO procedure
e.g. SOLO: EDIT NOTE
Oops... you can't LIST built in SOLO procedures (like NOTE, etc.)
number of occurrences: 1
syntactic category: none
conceptual category: system procedure changes allowed
status: improve message
priority: high RTC:3 EST: 00:30

Change 56.1: Give error message no. 10 for this error.

This error should have received error number 10 (which had 0 occurrences):

SOLO: EDIT NOTE
Oops... you can't EDIT a built-in SOLO procedure (like NOTE, etc.)

This occurred only 1 time, and would be handled by this change in choice of error message.

BEST COPY AVAILABLE

t57: Synonym corrector works
e.g. SOLO: END RUN
Oops... the format for BYE is BYE (and does a logout)
number of occurrences: 2
syntactic category: wrong command used
conceptual category: none
status: ok
priority: na RTC: 1 EST: 00:00

Change 57.1: Turn off synoym feature of spell corrector (see change 13.1)

Turn off the synonym, feature, as mentioned earlier (change 13.1). These two successes don't warrant the existence of the synonym matcher with its' large flaws.

t58) Attempt to NOTE an assertion that already exists in the database
e.g. NOTE FIDO---ISA--->DOG
OK... (I already knew FIDO---ISA--->DOG)
number of occurrences: 322
syntactic category: none
conceptual category: over writing existing assertions is legal
status: improved message
priority: high

t59) Attempt to FORGET an assertion that does not exist in the data base
e.g. SOLO: FORGET FIDO ISA PIG
ok... (I didn't know FIDO ISA PIG)
number of occurrences: 116
syntactic category: none
conceptual category: believes assertion exists in database when it does not
status: Improved message
priority: high
total number of occurrences: 438 RTC: 2 EST: 01:13:00

Change 58.1: Change "OK..." to "Oops..." to make an error explicit

Change response message from: OK... (I already knew FIDO ISA DOG)

to

Oops... (I already knew FIDO ISA DOG)

BEST COPY AVAILABLE

The rationale for this change rests in the fact that the concept of "overwriting" is common to beginners, but doesn't hold in SOLO. One cannot merely NOTE over an assertion. It must be forgotten, then re-NOTE.d. In actuality, overwriting is an error and should be acknowledged as one with an "Oops..." instead of the "OK..." which is used to acknowledge a legal SOLO command being carried out. If a user is attempting to remove an assertion which doesn't exist, his/her control flow could be wrong, or commands could be failing to be performed.

Although this might seem to be a trivial point, it leads to a larger set of possible errors, which should be flagged with a standardized error flag, namely "Oops".

t60) Attempt to create a procedure which already exists
e.g. SOLO: TO PRAISE /X/
Oops... I already have a procedure called 'PRAISE' /X/
number of occurrences: 63
syntactic category: possibly wrong call to the editor
conceptual category: creating and editing a file are synonymons
status: ok
priority: na

Change: none

The 63 occurrences of this type were handled adequately by this error message at this point. Change 61.1 handles specific possible intentions of users.

BEST COPY AVAILABLE

t61) A positive response to redefining procedures
e.g. would you like to redefine it now? (yes or no): Yes
Go ahead, but remember that any procedures which use the old 'Praise' /X/ will have to be redefined if they are to work properly
number of occurrences: 23
syntactic category: none
conceptual category: may think this is the same as editing
status: improve message
priority: high

t62) Negative confirmation to redefining procedure
e.g. I already have a procedure called 'PRAISE' /X/
Do you want to redefine it now? (Yes or No): NO
number of occurrences: 19
syntactic category: none
conceptual category: possible misunderstanding of message
status: improve message
priority: high

t63) No response to a confirmation message
e.g. Do you want to redefine it now? (Yes or No):
Oops... I already have a procedure called 'PRAISE' /X/
Do you want to redefine it now? (Yes or No):
Number of occurrences: 25
syntactic category: no arg given
conceptual category: not understanding error
status: improve message
priority: high
total number of occurrences: 67 RTC: 3 ESTL 33:30

Change 61.1: Disambiguate the meaning of "redefine" in message nos. 32, 33 and 35.

Improve message by disambiguating the wording of this message. "Redefining" is ambiguous to the novice user. Does it mean giving a new procedure line and parameters, rewriting the whole thing, or simply editing? The proposed change to error messages number 32 and 33 would be from using the word "redefine it" to "erase the old version of X and rewrite the entire procedure".

This change would effect a total of 46 cases, lending increased understanding as to the decision at hand, that of redefining the procedure completely or not.

For example, the current message

BEST COPY AVAILABLE

"Would you like to redefine it now?"

the revised message would read

"Would you like to erase the old version of X and rewrite the entire procedure now?".

t64) Use of ' instead of " to delimit a string
e.g. Print 'HELLO WORLD"
[' = "]
Please use " to end a quoted-string, and not '
Also check use of SHIFT key
number of occurrences: 11
syntactic category: wrong delimiter
conceptual category: none
status: ok
priority: na

Change: NONE

This correction appears to work well, picking out only occurrences of ' which appear to be delimiters, and leaving alone those embedded in text, and appearing to be part of contractions. The current message clearly handles all 11 occurrences of this error, but the improved line should be reprinted with the added changes.

t65) student successfully concludes the creation of a procedure (using TO)
e.g. ....: DONE
ok... I now know how to 'PRAISE' /X/ SOLO:
number of occurrences: 274
syntactic category: none
conceptual category: none
status: ok
priority: na

BEST COPY AVAILABLE

Change: NONE

The message is direct and echoes the procedure name and arguments after the DONE command was given in edit. It covers all 274 occurrences.

Change 65.1: Remove fatal bug from DONE format error

A special case of error involving the DONE command has a nasty bug. If the user makes a type ahead error, the following occurs.

```
...: DONEK LIST FOO
Oops... the format for DONE is:
  DONE
SOLO:

number of occurrences: 1
priority: High   RTC: 5 EST: 15:00
```

What has occurred is that the error was trapped, but could have been given to the student as a legal command. Instead, currently the implementation is such that it gives the error message and then ends the edit without updating the changes that have been made during that editing session. The changes are lost. This should be changed immediately. The format error should be included, and the DONE command, carried out for the student, but the editing changes should be updated.

priority: High

t66) Student successfully concludes the editing of a procedure using EDIT

```
e.g. ...: DONE
OK... I have redefined how to PRAISE /X/ SOLO:
number of occurrences: 339
     syntactic category: none
     conceptual category: none
     status: ok
     priority: na
```

BEST COPY AVAILABLE

Change: NONE

The special change for the previous type (65.1 special change) is recommended. Otherwise, this message is clear and covers all 339 cases in which it occurred.

t67) Missing control flow statement in conditional
e.g. 10A If Present: Print "I can't count higher than this"
(Please specify EXIT, CONTINUE, or NEXT CASE after the ; below)
number of occurrences: 107
syntactic category: missing statement
conceptual category: absent flow of control
status: improve message
priority: low RTC: 1 EST: 00:00:00

Change 67.1: Add separate control flow prompt messages for FOR EACH and CHECK mode

Improve message by having separate messages for check in FOR EACH and CHECK in other contexts. The option NEXT CASE as a control flow argument is used only in a very specific case, which most SOLO users never get to. It should be prompted for only where it is appropriate. This cuts down on the amount of information presented to the user who is still working out the uses of CONTINUE and EXIT, and doesn't need to worry about an obscure third option. Examples of this difference follows:

```
...: 1 CHECK FIDO---LIKES--->BEER
..... 1A If Present: PRINT "HE LIKES IT"
(Please specify EXIT or CONTINUE after the ; below)
.... 1A If Present: PRINT "HE LIKES IT";
```

This message exists (message number 47) but isn't used.

BEST COPY AVAILABLE

```
....1 1 FOR EACH FIDO---LIKES--->?
..... 1A CHECK * contains alcohol
..... 1AA If Present: NOTE FIDO MUSTBE BOOZER
(Please specify EXIT, CONTINUE, or NEXT CASE after the ; below)
..... 1AA If Present: NOTE FIDO MUSTBE BOOZER;
```

t70) Student forgets line number while in edit mode
e.g. TO PRAISE /X/
PRINT "I THINK THAT" /X/
Oops... you forgot the line number
number of occurrences: 116
syntactic category; missing line number
conceptual category: none
status: ok
priority: na

Changes: NONE

This message covered the 116 occurrences of the error. The only note here is that this error would disappear if the new implementation were micro processor based with a nicely designed screen editor.

t71) Student attempts to use NEXT CASE as control flow in a CHECK statement that isn't within a FOR EACH
e.g. 20A If Present: NEXT CASE
(NEXT CASE ONLY allowed in FOR EACH construct)
number of occurrences: 4
syntactic category: wrong control statement for construct
conceptual category: confusion about concepts and actions of different constructs
status: improve message
priority: medium RTC: 3 EST: 02:00

Change 71.1: trap and give specific message for wrong control flow word

A specific message could be triggered which would specify the error:

BEST COPY AVAILABLE

```
...20A If Present: NEXT CASE
 Oops... NEXT CASE not allowed in this occurrence of a CHECK
 statement.
 (please specify EXIT or CONTINUE after the : below)
...20A If Present: ;
```

This change would be in conjunction with change 67.1 to clarify the possible cases where NEXT CASE is an appropriate statement.

t72) Line number out of legal range

```
e.g. ...: Ø Print "THIS IS THE FIRST LINE"

  or

...: 15Ø Print "THIS IS FUN"
 Oops... You forgot the line number
number of occurrences: 27
     syntactic category: out of legal bounds of a variable
     conceptual category: none
     status: new message
     priority: high   RTC: 3   EST: 13:30
```

Change 72.1: Give new message for line number out of legal range

A new message is urgently needed. This message is inappropriate for the error. A more explicit message would be as follows:

```
...: 15Ø Print "THIS IS FUN"

Oops..: the line number you used in the last line was out of
the legal range.  You can only use whole numbers between 1 and
100.
```

This change would clarify all 27 of the occurrences of this error.

BEST COPY AVAILABLE

t73) Reference to a variable which hasn't been assigned a value yet
SOLO: TO FOO
e.g. ...: 1 PRINT "THE ANSWER IS" *A
number of occurrences: 56
syntactic category: none
conceptual category: mixing up variable and parameter concepts, or scope of variable confusion
status: trap
priority: high RTC: 4 EST: 40:00

Change 73.1: Trap unbound variables before run time

This error shows up as a run time error but it could be intelligently pre-empted in all cases by use of an intelligent demon at edit time. This demon should look for uses of * variables which haven't been filled yet during the procedure. If a line is added to a procedure during an editing session, and that line contains a * variable that is unreferenced, the user should be warned:

```
...: 1  PRINT *
Oops... the variable * has no value (contents) at this point
(type HELP * for more information)
...:
```

There has been no previous reference to the ? variable in this procedure, and a clever demon could keep track of that fact, and short circuit the conceptual bug before RUN-time. These errors are very costly time-wise to debug later. A debugging session for an error of this type involves.

- finishing the edit and running the procedure
- getting the proper flow of control to exhibit the error and receive the error message
- interpreting the error message
- figuring out the problem conceputally
- possibly turning on the TRACE
- isolating the error
- listing the procedure to see the line needing change
- editing the procedure to change the line
- finishing editing and re-running

This time could be estimated at probably 2 minutes of an experienced user, but for a beginner might mean ½ hour or more of connect time, much of which could have been pre-empted.

This error occurred 56 times at runtime.

t74) Student attempts to create a procedure with the same name as a built in SOLO procedure
e.g. EDIT LIST
number of occurrences: 1
syntactic category: none
conceptual category: reserved name violation
status: improve
priority: low RTC: 3 EST: 00:30

Change 74.1: Improve message by giving message number 10 in this case. Message number 10, although never used, appears to have been added to the message list to handle exactly this occurrence, and seems to do this more explicitly than the existing message. The error is currently handled thus:

```
SOLO: EDIT LIST
Oops... there's already a built-in SOLO procedure
called LIST
```

The same error, handles with message number 10 would be:

BEST COPY AVAILABLE

BEST COPY AVAILABLE

```
SOLO: EDIT LIST
 Oops... you can't EDIT built-in SOLO procedures
        (like NOTE, etc).
```

This error only occurred 1 time, and was handled relatively well by the existing system, hence the low priority.

t75) Addition of sublines (e.g. 2A) where one is not legal

```
 e.g. ...: 2A PRINT "THIS IS THE NEXT STEP"
 (line 2 doesn't require a substep)
 (2A = 2 A) Oops the format for A is:
number of occurrences: 40
     syntactic category: subline not legal
     conceptual category: subline construct legal in all
      places or made an error editing
     status: trap
     priority: High  RTC:4  EST: 03:20:00
```

Change 75.1: Give explicit message about illegal subline; use subline demon

This error owes part of its occurrences to the non-transparency of the SOLO editor. If the editor made explicit the ramifications of delete commands, the number of occurrences of this type would be reduced. However, there are simply cases where users believe that they are allowed to add sublines at will. A demon which looked for illegal sublines could give an intelligent message about the nature of the error, instead of ending up with a procedure error as the unintelligent spelling corrector now produces, e.g.

```
...: 2A PRINT "THIS IS IT"
Oops... you tried to add the subline 2A where it is not legal
(Type HELP for more information)
```

Extended HELP: You tried to add line number 2A, but this is not legal, either because:

a) line 2 does not use sublines (only lines with CHECK or FOR EACH constructs require sublines) or

b) line 2 does not exist or was inadvertantly removed during one of your edits.

Perhaps you should type 'DONE' to finish editing, then use LIST to see which problem you have.

BEST COPY AVAILABLE

Some of the 40 occurrences of this type only appear later as routine errors, which makes them extremely difficult to find and change (as is detailed in change 73.1) and combined with a more transparent editor (one which states all changes to a procedure and their ramifications) would trap all occurrences of this error at edit time.

t76) Attempts multiple nodes from a single node-relation pair
While in "CHECK mode"
e.g. NOTE 2ADD---6--->78
Oops... 2ADD---6--->9 already
number of occurrences: 183
syntactic category: none
conceptual category: Overwriting possible or confusion about uniqueness of triples
status: improve
priority: low RTC:2 EST: 30:30

Change 76.1: Add a line to message no 102 for BEGINNER modes

This change is minor, and consists of adding a line for the existing error message in BEGINNER mode or an extended HELP message to state the concept behind the error, e.g.

SOLO: NOTE 2ADD---6---8
Oops... 2ADD---6--->9 already

(the node relation pair 2ADD---6 only have one node coming from it, and that is already in your data base, namely 8. If you wish to change the database, you must first FORGET the original triple then NOTE the new triple).

This message occurred 183 times and was relatively well handled by the existing message, but could be improved.

BEST COPY AVAILABLE

t77)Used O instead of Ø or l instead of 1
e.g. 20 FORGET FIDO ISA DOG
(20 = 2 0)
Oops... the format for O is...
number of occurrences: 2
syntactic category: wrong symbol used
conceptual category: none
status: trap
priority: high RTC: 5 EST: 30:00

Change 77.1: Trap and substitute O/Ø and l/1 errors

This error is particularly common among beginners who have some typing experience. The error itself is nearly invisible, as the characters appear the same on many displays. The error appears first at login time, and should be trapped there, but at other times as well (i.e. line numbers). SOLO could have a demon for login time and line number input time in the editor to look for and make such substitutions for the user:

Student number please: C0123812
Oops... you used O (the letter) instead of using Ø (zero, the number)
You used l (the letter) instead of 1 (one, the number)

This will cause confusion later, so make sure you use only the actual numbers when you wish to type numbers.

and/or

They may look almost the same, but SOLO reads them only as they are typed, an e is an e and a 1 is a 1)

Actual counts of this error during login were not available, however this figure will be included in future analyes. This error occurred only 2 times in line numbers, but each time had serious side effects of deleting lines that should have been left intact.

BEST COPY AVAILABLE

t78) Type in student number after already logging in
e.g. SOLO: BØ1191X
Oops... I don't have a procedure called BØ1191X
number of occurrences: 11
syntactic category: not a legal procedure name
conceptual category: either type-ahead or doesn't believe that s/he has successfully logged in.
status: trap
priority: medium RTC: 2 EST: 01:50

Change 78.1: Trap student ID number after successful login

Have a demon to deal with this case of type-ahead or confusion about the login procedure. If the student number of the person already logged on is found at the top level, then the following message would be appropriate:

SOLO: CØ123812
Oops... that looks like your student number. You have already successfully logged into SOLO and it is now ready for your commands (e.g. DESCRIBE NOTE, FORGET LIST etc.)

This message would apply to 11 occurrences of errors of this type.

With improved friendliness and intelligence.

t79) Attempt to have an interactive program
e.g.
SOLO: SUBTRACT

OK...

THE ANSWER IS 109

DO YOU WANT TO TRY ANOTHER PROBLEM?

SOLO: YES
Oops... I don't have a procedure called YES
number of occurrences: 2
syntactic category: thinks I/O is possible with running program
status: ok
priority: na

Change: None

It would be too difficult to try and trap this conceptual error. The error is generally one which is very high level, and the mark of a sophisticated, creative user who can most probably figure out the error on his/her own.

It only occurred 2 times.

BEST COPY AVAILABLE

t80) User uses a synoym for a SOLO procedure for which no synonym exists in the spelling corrector
e.g. DEFINE SUBCHECK (instead of TO)
ADD FIDO ISA DOG (instead of NOTE)
number of occurrences: 4
syntactic category: illegal procedure name
conceptual category: use of synonym instead of legal procedure name
status: ok
priority: na RTC: 1 EST: 00:00

Change 80.1: Turn off synonym feature of corrector (see change 13.1)

The same as 13.1. The number of different possible synonyms which overlap with logical, probable student defined procedure names is too great, and hence runs the risk of error 13.

t81) Error message while in "CAT" option
e.g. (procedures refer to other procedures which have yet to be defined)
Oops... I don't have a procedure called X
number of occurrences: 121
syntactic category: none
conceptual category: undefined procedures still referenced in other procedures
status: improve message
priority: low RTC:1 EST: 00:00

Change 81.1: Improve and clarify message in CAT option

Change message from the existing one to something short like:

"Procedure(s) which don't exist that this procedure uses:

FOO /ME/

ZOOM /X/"

This would save time during the printout and entrance point for which this message was added originally.

This message occurred 121 times.

BEST COPY AVAILABLE

t82) Format error in BRIEF mode
e.g. NOTE FIDO I
number of occurrences: 9
syntactic category: format error, not enough Args
conceptual category: none
status: ok
priority: na

Change: NONE

All nine occurrences were handled as the user desired to have them handled (i.e. briefly).

t83) Spelling corrector makes correct change to input line
e.g. "10PRINT = 10 PRINT"
number of occurrences: 271
syntactic category: missing space delimiter
conceptual category: none
status: ok
priority: na

t84) Spell corrector gives the wrong split to a user input line
e.g. POSSIBLYHAS = PO SSIBLYHAS
or
10ASSIGN = 10A SSIGN
number of occurrences: 8
syntactic category: missing space
conceptual category: none
status: trapable
priority: medium

BEST COPY AVAILABLE

t85) Spelling checker asks for confirmation on a plausible (word ± 2 chars) spelling change; Accepted by user
e.g. when you typed MAYHAVE did you mean MAYHAVE (yes or no): yes
number of occurrences: 130
syntactic category: mispell
conceptual category: none
status: ok
priority: na

t86) Spelling checker asks for confirmation on a plausible change (word ± 2 chars); Rejected by user
e.g. when you typed JEAN did you mean JOAN
number of occurrences: 49
syntactic category: none
conceptual category: none
status: ok
priority: na

t87) Spelling checker does a silly number check
e.g. When you typed 14 did you mean 1
number of occurrences accepted: 2
number of occurrences rejected: 93
syntactic category: in accepted case above; mispell
conceptual category: none
status: trap
priority: high

t88) Spelling checker tries a ridiculous possible match (word +/- >2 chars)
e.g. When you typed JUMPROPE did you mean J?
number of occurrences Accepted: 4
number of occurrences Rejected: 93
syntactic category: in accepted case above; misspell
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

Changes: 83-88

These types all deal with the spelling corrector, which has been dealt with earlier (see section 4.1) in regards to recommendations for specific changes.

BEST COPY AVAILABLE

4.3) GENERAL CHANGES

The following changes were gathered from general observations, anecdotes and scribbled notes while sitting at the terminal and working with students. Several of these changes are quite important and have been appropriately labeled so in the Summary Pages in appendix A.

Change G1.1: Compile time warning about undefined procedures

When the student finishes editing s/he should be warned about the existence of any undefined procedures which that procedure invokes, e.g.,

```
SOLO: TO FOO /X/
 ...: 1 CHECK /X/ ISA PERSON
 .....1A If present: IDENTIFY /X/; EXIT
 .....1B If absent: REPLY; EXIT
 ...: DONE
(Warning... this procedure uses procedures:
          IDENTIFY /X/
          REPLY
                which you have not written yet)
Ok... I now know how to FOO /X/
SOLO:
```


BEST COPY AVAILABLE

Change G2.1: Deal with undefined parameters and variables

This change is partially dealt with in change 48.1, but is defined only for print statements there. The more general change calls for a demon to keep track of parameters and variable calls. If a person uses a parameter in the body of a procedure and that parameter has not been passed into the procedure through the title line of the procedure then that fact should be pointed out as an error as soon as possible, e.g.,

```
SOLO: TO FOO
...: 1 NOTE /X/ ISA GIVEN
Oops... you just used the parameter /X/ in the previous
        line, but you didn't include it in the title
        line of the procedure.
        (Type HELP for more information concerning this error)
...:
```

Conversly, if a person includes a parameter in the title line and that parameter is never used in the body of the procedure then that fact should be pointed out to the user when they finish

BEST COPY AVAILABLE

editing, e.g.,

```
SOLO: TO FOO /X/
...: 1 NOTE FIDO EATS BISCUITS
...: DONE
Warning... you used the parameter /X/ in the title line
           of the procedure FOO but you never used /X/
           anywhere in the body of the procedure. You
           might be wise to check out this situation.
           (Type HELP PARAMETER for more information)
Ok... I now know how to FOO /X/
SOLO:
```

Change G3.1: Improve runtime error messages

Runtime errors should give the procedure name and line number on which the error occurred. This would help the user pinpoint the error more quickly and save time normally spent activating the TRACE facilities, adding print statements and listing procedures to find the culprit.

Change G4.1: Add an interrupt key

Some kind of interrupt is needed to terminate a running SOLO procedure. It should be well hidden from the novice user, perhaps as a shift/control character.

Change G5.1: Mandatory non-numeral first character in procedure

BEST COPY AVAILABLE

names

Procedure names should be restricted to permit only alphabetic characters as the first character in the name. This would facilitate the implementation of a line number demon. There were no cases where numerals were used in the first position, but if this occurred a message could be presented, e.g.,

```
SOLO: TO 3TIMES /X/

Oops... sorry, but your procedure name must
        start with a letter. You can use numerals
        anywhere else in the name except as
        the first character.
(Type HELP for more inforamtion concerning this error)

SOLO: HELP

Ok...
```

The way SOLO was designed made it necessary to forbid people from using procedures which begin with numerals. This is because SOLO tries to be as helpful as possible and this constraint assists in making that possible.

Change G6.1: Make SOLO more transparent

A general suggestion, the specific cases of which are detailed in the "Type Related Changes" section of this paper), is to make the "SOLO machine" more transparent. This means making explicit all changes that SOLO makes to user input lines at the time those changes are made. For example, if SOLO throws away an illegal character from a string, the user should be told, and the resul-

BEST COPY AVAILABLE

tant line reprinted. Currently hidden changes often appear to lead to increases in later debugging time. The student must be made aware of the errors which s/he is making as they are made, whether during an edit or at top level.

Change G7.1: Change wait message in typing lesson

While in the typing lesson, if the user waits without inputting any lines for a certain length of time, the following message appears:

INPUT ERROR - You are wasting time -hurry up.

The tone of this message is totally uncalled for and works opposite to all the ideals of "friendly" design (and incidently existed for three years without bieng known to Marc Eisenstadt or any of the D303 course team). The message which should occur is message number one. This message warns that time has passed and directs the user to hit the RETURN key.

Change G8.1: Add a ":" after the subline prompt in FOR EACH

This change is for reasons of continutiy. All other prompts end

BEST COPY AVAILABLE

in a colon. The current FOR EACH subline prompt appears as:

.....2A

The revised version would appear as:

.....2A:

Change G9.1: Improve typing lesson

Currently the typing lesson which is offered to first time SOLO users is basically acceptable. It introduces students to typing and hitting the return key at the ends of an input line. Two major changes are suggestesd, however.

1) Lengthen typing lesson but intersperse confirmation type queries as to whether the person would like to have more practice. This not only helps the student who would like lots of practice, but it introduces users to the idea of "confirmation" and the typing of yes or no.

2) Change the example which deals with the use of the RUBOUT key. The current example is very confusing and could put a person off right from the start. It is suggested that some standard examples be given, then a short paragraph about using the RUBOUT key and its response be presented. Some examples should be given that let the user play with typing and backspacing, comparing

BEST COPY AVAILABLE

what s/he thought SOLO would get with what it actually received, e.g.

> Type in: MESSUP
>
> then type the rubout key several times. Before you hit the return key, see if you can predict what line you ended up with after all the rubouts were typed!

Several of these excersises could be given. The idea is to build a learning session that is flexible enough to meet the needs of a fast, agressive learner, and still be safe for a non-agressive learner. This is also a good chance to introduce the basic statement structure and get people used to paying attention to the details of syntax. This is done relatively well in the current implementation, but could be improved.

Change G10.1: Warn student about use of "/" where inappropriate

Currently if a user types in:

SOLO: PRAISE /FIDO/

SOLO simply throws out the "/"s and runs the procedure if it exists. This is the intelligent thing to do in most cases, however the student should be warned about the illegal use here. This could be a cue to a conceptual overgeneralization about parame-

BEST COPY AVAILABLE

ters. A message such as the following should be presented:

```
SOLO: PRAISE /FIDO/
   (The "/"s you just used are not necessary here.
    I'll remove them for you and continue. Type
    HELP PARAMETER for more information)
      PRAISE FIDO
OK...
```

Change G11.1: Explain ?X and *X in messages 55 and 17

Add a line or two which explains the use of X as a variable name, e.g.,

```
Oops... the format for FOR EACH is:
        FOR EACH CASE OF node---relation--->?X
 (Type HELP for more information)
...: HELP
Ok...
```

The X in that message stands for any letter you'd like to pick, for example: ?A, ?K or ?W . Read the last paragraph on page 69 of the primer for an explanation. For more information on the FOR EACH construct and how it works type:

```
HELP FOR EACH
 ...:
```


BEST COPY AVAILABLE

Change G12.1: Add --- and ---> in FOR EACH

Add the delimiters --- and ---> automatically in the printout of lines involving FOR EACH (the way it is currently done for CHECK, NOTE and FORGET).

Change G13.1: Eliminate the ";" on certain control lines

Remove the ';' which occurs before a CONTINUE or EXIT on a sub-line if no procedure is specified on that same line, e.g.,

.....1A If present: CONTINUE

Currently looks like this:

.....1A If present: ;CONTINUE

Change G14.1: Allow options to be preserved on logout

Allow all the set user options (SILENCE, etc.) to be preserved on logout so that they get restored to their most recent status on the next login, (with an appropriate reminding message).

BEST COPY AVAILABLE

5) DISCUSSION

The validity of all changes to SOLO, whether additions of demons and algorithm changes or the removal of a feature to speed up the system, must be assessed by asking three questions;

--How does it effect the system load

--How would it effect the pedogigical ideas behind the structure and teaching of SOLO

--How will it effect the growth of the learner

These questions deal with three aspects of Computer Aided Instruction: CPU and system reaction time, the environment and the user. The first is the reality of system load. This question will become less important as the use of single user micro processors becomes more prevalent, but is of high importance with time sharing facilities. What good is a teaching system if it consumes a large percentage of connect time waiting to respond, and slowing down all other users?

The second question deals with the effect of the change on the SOLO environment.

--Does the change improve the conceptual continuity of the environment?

--Would the change make it easier for a person to do what s/he desired to do?

--Would it save the user connnect time now wasted recovering

BEST COPY AVAILABLE

from "non-productive" errors?

The pre-emptive design ideal behind SOLO dictates that the designer attempt to pre-empt all user errors. Part of that ideal is disambiguation of conceptual errors implicit in the user input. Granted, "silly" errors do occur, but a large class of errors are conceptually generated, and possible culprit concepts can often be pinpointed, and corrected. By taking the blame for all errors the assumptions of efficient design fall entirely on the designer, and avoid expectations and assumptions about the user responsibility.

Finally, the effect of changes on the user must be be considered. Does a relatively complex "improvement" to the environment costing programmer hours in implemtation and added CPU time to run warrant its existence on the fact that it makes learning easier for some users in a relatively small, perhaps "trivial" set of circumstances? To answer this question an effect which is called the "Primary Math Book Trap" should be considered.

The "Primary Math Book Trap" can be found in all learning situations. As a definition, think of the situation of a child looking at the first chapter of his/her new math book on the first day of the new school year. To many, on glancing through that first chapter, it appears as an insurmountable hurdle. The symbols are new and strange, the unpronouceable words and the sheer length of the chapter can present a menacing, "impossible"

BEST COPY AVAILABLE

learning task. However, with quality teaching the child is helped through this first "maze" and the rest of the chapters over the course of the year. Yet at each chapter the effect can begin again, and the previous chapter is seen as having been "so easy" or "trivial", compared with what must be learned at present. The tendency, in hindsight, is to forget the fact that at the start of each chapter the new pages were ominous, mysterious and generally incomprehensible.

The same is true, to some extent, of computer people and software designers. We have a tendency to forget the perspective which the novice computer learner brings to the learning session, (particulary the non-math oriented social science student). This perspective should be considered before any label such as "trivial" is applied to a class of errors or a proposed change to the system.

SOLO currently appears to be performing admirably in teaching the principles of A.I. at a distance. This "hands on" computing experience is critical to the A.I. emphasis of the D303 course. The intelligent user aids and error handling reviewed in this paper are an integral part of the "friendly" user environment. It is this "friendliness" which appears to be making SOLO as popular and successful as it is currently.

The decision concerning implementation of changes (whether

BEST COPY AVAILABLE

the removal or improvement of existing features), involves a weighing of utilities and an examination of the goals held by the university. My strong recommendation is for the retention and improvement of the user aids and general environment. This is consistent both with the goals of the institution and the desire to be truly cost effective. Not only will the proposed changes make SOLO a better teaching aid, but they will also save an estimated 7.5-15.0% of the total user connect time which is currently wasted.

If SOLO's pedogogical effectiveness can be improved and connect time can be reduced by implementing changes, then the university stands to benefit twofold.

BEST COPY AVAILABLE

6) REFERENCES

Du Boulay, B., O'Shea, T., Monk, J. "The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices". D.A.I. Research Paper No. 133, 1980.

Eisenstadt, M. "Cognitive Psychology: Artificial Intelligence Project". Open University Press, Milton Keynes., 1978.

Feurzig, Papert, Bloom, Grant, and Soloman. Programming languages as conceptual frameworks for teaching mathematics. Report no. 1889, Bolt Beranek and Newman, Inc., Cambridge, Mass., 1969.

Teitelman, W. "Interlisp reference manual." Xerox PARC, Palo Alto, 1974

BEST COPY AVAILABLE

Appendix A: Summary Pages

The following pages contain a summary in table form of all the changes, their requirements and benefits. The times are all approximate due to absence of time information being stored with the data (as is detailed in section 2: Methods, page 20.

Total estimated time saved from:

Changes to the HELP OPTIONS (see page 116)		9:30:50
Type related changes (see pages 117-122)		39:29:10
General Changes (see page 123)		10:00:00
	Total:	59:20:00

Total number of connect hours for 1979 D303 Summer school: 1144:00:00

Analysis on and estimated 70% of this total

70% of 1144 = 800 hrs.

Maximally conservative estimate of percent of hours saved = 60/800

= 7.5 %

Plausible estimate of percent of hours saved = 15.0%

Possible total number of hours saved for the entire summer: 7.5% of 1144 = 86

BEST COPY AVAILABLE

Summary sheet for HELP OPTION changes

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	time saved	added CPU time?
S&L	4.1a Restructure and improve the spelling corrector	2	216	low	00:36:00	0
L	4.1b Change the user access to the 'HELP OPTIONS'	2	389	low	01:04:50	+
S	4.1c Improve the tracing facilities	4	98	high	08:10:00	+
L	4.1d Review the remaining options	1	273	med.	00:00:00	+
L	4.1e Rename some options	1	389	med.	00:00:00	0
L	4.1f Possibly add a "Very helpful", verbose beginners mode	?	?	high	????????	+
L	4.1g Add a lesson in HELP usage	?	?	high	????????	+

Total estimated time saved from changes to the HELP OPTIONS 09:50:50

BEST COPY AVAILABLE

Summary sheet for Type Related changes (1 of 6)

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
L	1.1 Intelligently handle 'too many arguments' to NOTE and FORGET					
	1.2 Intelligently handle 'too many arguments' to LIST, DESCRIBE and EDIT	2	52	med.	00:08:40	+
L	2.1 Link spell corrector to argument counting demon	2	20	low	00:03:20	0
L	3.1 Add 'pair crunching' and argument counting demons to spell corrector	2	44	low	00:07:20	+
L	4.1 Intelligent parsing of conditional lines	3	21	med	00:21:00	+
L	5.1 Add disambiguating message for conceptual error of 'too few arguments'	2	84	low	00:14:00	+
L	6.1 Add "node/relation-splitter" with argument counting demon	2	61	low	00:10:10	+
	7 NO CHANGE	-	67	---	--------	0
S	8.1 Add input cut-off to reduce type-ahead problems	3	99	med	00:49:30	+
S	9.1 Trap special format error and give clarifying message	3	19	high	00:09:30	+
S	10.1 Intelligently handle illegal characters using argument counting demon	2	43	low	00:07:10	0
S	11.1 Add special message about name length limit or extend limit	4	36	high	03:00:00	0
S	13.1 Turn off synonym feature of spell corrector	5	26	high	06:30:00	0
				total	11:40:40	

123

124

BEST COPY AVAILABLE

Summary sheet for Type Related changes (2 of 6)

S(hort)or L(ong) term	change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
L	14.1	Trap and clarify concept behind over extended use of "wild card"	4	7	high	00:35:00	0
L	17.1	Restructure HELP access (see section 4.1b)	3	8	med	00:04:00	0
L	18a.1	Special message for EDIT mode error	3	42	high	00:21:00	0
L	18b.1	Specila message for top level mode errror	3	21	high	00:10:30	0
L	19.1	Explain that FORGET can't work on on a procedure	2	1	med	00:00:10	+
L	20.1	Trap and explain illegality of attempted concatenation	5	4	low	01:00:00	+
	21	NO CHANGE	-	14	---	--------	0
S	22.1	Add missing slash to parameter name in title line	2	3	low	00:00:30	+
S	23.1	SOLO should accept To or parameters with procedure names	3	25	high	00:12:30	?
L	24.1	Trap and explain 'EDIT node name' error	3	3	med	00:01:30	+
	25	NO CHANGE	-	115	---	--------	0
L	26.1	Prompt for, or give obvious forgotten SOLO procedure name	2	60	low	00:10:00	+
S	27.1	Trap and explain single number error	2	13	high	00:02:10	0
	28	NO CHANGE	-	230	---	--------	
					total	02:37:20	

BEST COPY AVAILABLE

Summary sheet for Type Related changes (3 of 6)

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
L	29.1 Trap and explain data/procedure confusion 30.1 Trap and explain procedure/data confusion	3	34	high	00:17:00	+
	31 NO CHANGE	-	2	---	--------	0
S	32.1 Turn on spelling corrector in fancy features	2	93	low	00:15:30	+
	33 NO CHANGE	-	12	---	--------	0
L	34.1 Trap and disambiguate "*"/"?" confusion in lines	3	13	med	00:06:30	0
S	35.1 Trap special case of FOR EACH error and explain	3	32	high	00:16:00	0
L	36.1 Restructure HELP OPTIONS (see section 4.1b)	2	2	med.	00:00:20	0
L	37.1 Trap and explain 'wrong access' error	3	2	med.	00:01:00	0
	38 NO CHANGE	-	64	---	--------	0
	39 NO CHANGE	-	2	---	--------	0
	40 NO CHANGE	-	1	---	--------	0
	41 NO CHANGE	-	21	---	--------	0
	42 NO CHANGE	-	3	---	--------	0
	43 NO CHANGE	-	4	---	--------	0
	44 NO CHANGE	-	2	---	--------	0
L	46.1 Trap and explain illegal control flow statements	4	82	high	06:50:00	+
	-6-			total	07:46:20	

BEST COPY AVAILABLE

Summary sheet for Type Related changes (4 of 6)

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
L	47.1 Trap and correct delimiter problems	3	22	high	00:11:00	+
L	48.1 Trap and disambiguate parameter, variable and value problems in PRINT	4	36	high	03:00:00	+
L	49.1 Trap and explain illegality of multiple procedure calls on a line	4	14	high	01:10:00	+
L	50.1 Positive response to a confirmation statement 51.1 Negative response to a confirmation statement	2	517	low	01:26:10	-
S	52.1 Expand message to clarify #A/A confusion in parameter list	2	2	med.	00:00:20	0
S	53.1 Add student defined parameter names to message no. 11	2	97	low	00:16:10	0
	54 NO CHANGE	-	15	---	--------	0
L	55.1 Trap specific case of listing HELP OPTIONS and pre-empt	3	1	med.	00:00:30	+
S	56.1 Give error message no.10 for this error	3	1	high	00:00:30	0
S	57.1 Turn off synonym feature of spelling spelling corrector (number of times feature worked correctly)	1	2	high	00:00:00	-
S	58.1 Change "OK..." to "Oops..." to make error explicit	2	438	high	01:13:00	0
	60 NO CHANGE	-	63	---	--------	0
				total	07:17:40	

BEST COPY AVAILABLE

Summary sheet for Type Related changes (5 of 6)

S(hort) or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
S	61.1 Disambiguate the meaning of "redefine" in messages nos. 32, 33 and 35	3	67	high	00:33:00	0
	64 NO CHANGE	-	11	---	--------	0
	65 NO CHANGE	-	274	---	--------	0
S	65.1 Remove fatal bug from DONE format error	4	1	high	00:15:00	0
	66 NO CHANGE	-	339	---	--------	0
S	67.1 Add seperate flow of control prompt for FOR EACH and CHECK modes	1	107	low	00:00:00	0
	70 NO CHANGE	-	116	---	--------	0
L	71.1 Trap and give specific message for wrong control flow word	3	4	high	00:02:00	+
S	72.1 Give new message for line number out of range	3	27	high	00:13:30	+
L	73.1 Trap unbound variables before run time	4	56	high	04:40:00	+
S	74.1 Improve message by giving message no. 10 in this case	3	1	low	00:00:30	0
L	75.1 Give specific message about illegal subline; use subline demon	4	40	high	03:20:20	+
S	76.1 Add a line to message no. 102 for beginner mode	2	183	med	00:30:30	0
S	77.1 Add disambiguating message for conceptual error of 'too few arguments'	5	2	high	00:30:00	+
				total	10:04:50	

131

132

BEST COPY AVAILABLE

Summary sheet for Type Related changes (6 of 6)

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	est. time savings	Added CPU time?
L	78.1 Trap student I.D. number after successful login	2	11	low	00:01:50	+
	79 NO CHANGE	-	2	---	--------	0
S	80.1 Turn off synonym feature of corrector problems (this synonym doesn't exist)	1	4	none	00:00:00	0
S	81.1 Improve and clarify message in CAT option	1	211	med.	00:00:00	0
	82 NO CHANGE	-	9	---	--------	0

total 00:01:50

totals for:

page 1	11:40:40
page 2	02:37:20
page 3	07:46:20
page 4	07:17:40
page 5	10:04:50
page 6	00:01:50

Maximally conservative estimated saved time for the Type Related changes: 39:28:40

BEST COPY AVAILABLE

Summary sheet for the General Changes section

S(hort)or L(ong) term change	Number and Name of Change	Recovery time class	no. of occur.	educ. benefit	time saved	added CPU time?
S	G1.1 Compile time warning about undefined procedures	4	??	med.	????????	+
L	G2.1 Deal with undefined parameters and variables	4	??	high	????????	+
S	G3.1 Improve runtime error messages	3	??	high	????????	+
S	G4.1 Add an interrupt key	3	??	high	????????	0
L	G5.1 Mandatory non-numerical first character in procedure names	3	??	med.	????????	-
S	G6.1 Make SOLO more transparent	4	??	high	????????	0
S	G7.1 Change wait message in typing lesson	1	??	high	????????	0
S	G8.1 Add a ":" after the subline prompt in FOR EACH statement	1	??	low	????????	0
L	G9.1 Improve typing lesson	1	??	med.	????????	0
S	G10.1 Warn student about the use of "/" where inappropriate	2	??	med.	????????	+
S	G11.1 Explain ?X and *X in messages 55 and 17	2	??	med.	????????	0
S	G12.1 Add --- and ---> in FOR EACH	1	??	low	????????	0
S	G13.1 Eliminate the ";" on certain control lines	1	??	low	????????	0
S	G14.1 Allow options to be preserved on logout	3	??	high	????????	0

135

136

It is safe to assume that these changes would save at least a total of: 10:00:00

BEST COPY AVAILABLE

Appendix B: Breakdown by Types

The following pages cover each type of error which occurred in the data, and the breakdown of the type by the number of occurrences of each message number. The format of these tables is explained on page 23 of the methods section (section 3).

BEST COPY AVAILABLE

Appendix B: Index by "Types"

Type 1: Too many arguments to a procedure	Fraction of total in message index
m18: the format for NOTE is:	19/156 *
m19: the format for DESCRIBE is:	12/24
m20: the format for FORGET is:	1/42
m22: the format for LIST is:	1/42
m23: the format for EDIT is:	2/22
m25: the format for CHECK is:	3/44
m43: Wrong format	7/9
m101: type options here:	6/50
Total number of occurrences of this type:	52

Type 2: Spell corrector fault giving more than allowed number of arguements	Fraction of total in message index
m14: the format for user defined procedure is:	5/191
m18: the format for NOTE is:	2/156
m19: the format for DESCRIBE is:	2/24
m20: the format for FORGET is:	4/42
m55: the format for FOR EACH CASE OF is:	7/59
Total number of occurrences of this type:	20

Type 3: The user adds an unwanted space to the node/relation or procedure name after showing that s/he understands the format.	Fraction of total in message index
m4: you typed in too much	6/40
m18: the format for NOTE is:	19/156
m19: the format for DESCRIBE is:	1/24
m20: the format for FORGET is:	2/42
m22: the format for LIST is:	4/24
m24: the format for TO is:	6/33
m25: the format for CHECK is:	4/44
m29: I have no description of	1/148
m97: Sorry COPY Command Disabled	1/6
Total number of occurrences of this type:	44

*See p. 22 for an explanation of these numbers

BEST COPY AVAILABLE

Type 4: Flow of control statement taken as an arguement	Fraction of total in message index
m5: I don't have a procedure called (user defined)	13/369
m14: the format for (user defined)	4/191
m18: the format for NOTE is:	4/156
m19: the format for DESCRIBE is:	8/24
m20: the format for FORGET is:	4/42
m25: the format for CHECK is:	1/44
m37: Left Hand Quotes (") Assumed	4/155
m38: Right Hand Quotes (") Assumed	4/146
Total number of occurrences of this type:	42

Type 5: Too few arguements given to a procedure	Fraction of total in message index
m14: the format for user defined procedure is:	19/191
m18: the format for NOTE is:	35/156
m19: the format for DESCRIBE is:	0/24
m20: the format for FORGET is:	20/42
m25: the format for CHECK is:	10/44
Total number of occurrences of this type:	84

Type 6: Too few args because of missing space between node and relation name	Fraction of total in message index
m14: the format for user defined procedure is:	30/191
m18: the format for NOTE is:	22/156
m19: the format for DESCRIBE is:	1/24
m20: the format for FORGET is:	5/42
m25: the format for CHECK is:	3/44
Total number of occurrences of this type:	61

BEST COPY AVAILABLE

Type 7: No arguments given	Fraction of total in message index
m14: the format for user defined procedure is:	7/191
m15: the format for PRINT is:	3/4
m18: the format for NOTE is:	6/156
m20: the format for FORGET is:	3/42
m22: the format for LIST is:	1/24
m23: the format for EDIT is:	5/22
m25: the format for CHECK is:	3/44
m43: Wrong Format	2/9
m55: the format for FOR EACH CASE OF is:	3/59
m113 Do you want to select any more options	34/573
Total number of occurrences of this type:	67

Type 8: Errors due to type ahead problems	Fraction of total in message index
m4: you typed in too much	30/40
m14: the format for user defined procedure is:	4/191
m18: the format for NOTE is:	6/156
m20: the format for FORGET is:	0/42
m22: the format for LIST is:	2/24
m23: the format for EDIT is:	3/22
m24: the format for TO is:	3/33
m32: Would you like to redefine it now?	2/25
m33: Do you want to redefine it with	2/26
m36: (The Line Number was not necessary)...	9/28
m55: the format for FOR EACH CASE OF is:	2/59
m57: param1 param2 param3 (at most 3 parameters)	3/21
m63: I don't have a help option called	7/20
m76: Please specify EXIT, CONTINUE or NEXT CASE	3/126
m101: Type option here:	4/50
m113: Do you want to select any more options	19/573
Total number of occurrences of this type:	99

Type 9: Used "?" in a node in a NOTE or FORGET	Fraction of total in message index
m18: the format for NOTE is:	17/156
m20: the format for FORGET is:	1/42
Total number of occurrences of this type:	18

BEST COPY AVAILABLE

Type 10: Illegal characters in a node or relation names	Fraction of total in message index
m18: the format for NOTE is:	4/156
m20: the format for FORGET is:	0/42
m22: the format for LIST is:	1/24
m53: invalid procedure name	1/1
m97: Sorry COPY Command Disabled	1/6
Total number of occurrences of this type:	7

Type 11: Over legal limit for number of characters in a node or proceedure name	Fraction of total in message index
m4: you typed in too much	4/40
m14: the format for user defined procedure is:	8/191
m18: the format for NOTE is:	12/156
m20: the format for FORGET is:	0/42
m22: the format for LIST is:	2/24
m24: the format for TO is:	5/33
m25: the format for CHECK is:	2/44
m55: the format for FOR EACH CASE OF is:	3/59
Total number of occurrences of this type:	36

Type 12: Spurious control characters in node or relation names	Fraction of total in message index
m14: the format for (user defined)	5/191
m18: the format for NOTE is:	15/156
m20: the format for FORGET is:	0/42
m23: theformat for EDIT is:	1/22
m24: the format for TO is:	6/33
m25: the format for CHECK is:	6/44
m55: the format for FOR EACH CASE OF is:	2/59
m97: Sorry COPY Command Disabled	1/6
Total number of occurrences of this type:	36

BEST COPY AVAILABLE

Type 13: Steamroller effect by the synonym and spelling corrector	Fraction of total in message index
m15: the format for PRINT is:	1/4
m18: the format for NOTE is:	1/156
m20: the format for FORGET is:	1/42
m24: the format for TO is:	8/33
m25: the format for CHECK is:	4/44
m44: HELP not currently available for the above	1/14
m55: the format for FOR EACH CASE OF is:	1/59
m100: Spelling corrector	9/310
Total number of occurrences of this type:	26

NOTE: Types 15 & 16 were discarded

Type 14: Over extended use of wild card	Fraction of total in message index
m25: the format for CHECK is:	1/44
m55: the format for EACH CASE OF is:	7/59
Total number of occurrences of this type:	8

Type 17: Wrong access to help features	Fraction of total in message index
m37: Left Hand Quotes (") Assumed	1/155
m38: Right Hand Quotes (") Assumed	1/146
m44: HELP not currently available for the above	4/14
m101: Type option here:	2/50
Total number of occurrences of this type:	8

Type 18a: Mode confusion: student assumed s/he was in top level	Fraction of total in message index
m12: you forgot the line number	39/184
m25: the format for CHECK is:	3/44
Total number of occurrences of this type:	42

BEST COPY AVAILABLE

Type 18b: Mode confusion: student assumed s/he was in editor	Fraction of total in message index
m5: I don't have a procedure called (user defined)	10/369
m13: the variable *X has no value at this point	2/54
m36: (The Line Number was not necessary)	8/28
m89: you may only use 'FOR EACH' within a procedure definition	1/2
Total number of occurrences of this type:	21

Type 19: Attempt to apply a procedure to a proceedure	Fraction of total in message index
m24: the format for TO is:	1/33
Total number of occurrences of this type:	1

Type 20: Attempted concatenation of strings	Fraction of total in message index
m25: the format for CHECK is:	4/49
Total number of occurrences of this type:	4

Type 21: Use more tha three args to a user defined procedure	Fraction of total in message index
m14: the format for user defined procedure is:	12/191
m24: the format for TO is:	2/33
Total number of occurrences of this type:	14

Type 22: Forgot "/" in the procedure line during definition	Fraction of total in message index
m24: the format for TO is:	3/33
Total number of occurrences of this type:	3

BEST COPY AVAILABLE

Type 23a: Give procedure name with TO attached to it	Fraction of total in message index
m22: the format for LIST is:	5/24
m23: the format for EDIT is:	8/22
m97: Sorry COPY command disabled	1/6
Total number of occurrences of this type:	14

Type 23b: Gave procedure name with args included to a procedure	Fraction of total in message index
m22: the format for LIST is:	7/24
m97: Sorry COPY Command Disabled	2/6
Total number of occurrences of this type:	9

Type 24: Editing node names	Fraction of total in message index
m5: I don't have a procedure called (user defined)	3/369
Total number of occurrences of this type:	3

Type 25: Forgot " which SOLO then correctly added	Fraction of total in message index
m37: Left Hand Quotes (") Assumed	64/155
m36: Right Hand Quotes (") Assumed	51/146
Total number of occurrences of this type:	115

Type 26: Forgot to include an obvious SOLO procedure name	Fraction of total in message index
m5: I don't have a procedure called (user defined)	34/369
m36: (The Line Number was not necessary)...	8/28
m55: the format for FOR EACH CASE OF is:	12/59
m57: param1 param2 param3 (at most three parameters)	6/21
Total number of occurrences of this type:	60

BEST COPY AVAILABLE

Type 27: Give single number at top level	Fraction of total in message index
m5: I don't have a procedure called (user defined)	7/369
m36: (The Line Number was not necessary)...	8/28
Total number of occurrences of this type:	15

Type 28a: No referent	Fraction of total in message index
m5: I don't have a procedure called (user defined)	95/369
m29: I have no description of	106/148
Total number of occurrences of this type:	201

Type 28b: Describe a relation name with no referent	Fraction of total in message index
m29: I have no description of	9/148
Total number of occurrences of this type:	9

Type 29a: Attempt to use DESCRIBE on a procedure	Fraction of total in message index
m29: I have no description of	24/148
Total number of occurrences of this type:	24

Type 29b: Attempt to use LIST on a node	Fraction of total in message index
m5: I don't have a procedure called (user defined)	8/369
m23: the format for EDIT IS:	3/22
Total number of occurrences of this type:	11

BEST COPY AVAILABLE

Type 30: Use DESCRIBE as an inovation function	Fraction of total in message index
m5: I don't have a procedure called (user defined)	2/369
m11: there's already a built in SOLO procedure called	1/2
m29: I have no description of	6/148
Total number of occurrences of this type:	9

Type 31: Attribution of too much knowledge to a function	Fraction of total in message index
m29: I have no description of	2/148
Total number of occurrences of this type:	2

Type 32: Spelling error	Fraction of total in message index
m5: I don't have a procedure called (user defined)	16/369
m12: you forgot the line number	2/184
m37: Left Hand Quotes (") Assumed	11/155
m38: Right Hand Quotes (") Assumed	11/146
m44: HELP not currently available for the above	9/14
m57: param1 param2 param3 (at most 3 parameters)	7/21
m63: I don't have a HELP option called	9/20
m76: (Please specify EXIT, CONTINUE, or NEXT CASE...	4/126
m101: Type option here:	21/50
m113: Do you want to select any more options	3/573
Total number of occurrences of this type:	93

Type 33: Too many lines in a procedure	Fraction of total in message index
m54: Too many lines only 10 allowed	12/12
Total number of occurrences of this type:	12

BEST COPY AVAILABLE

Type 34: "=" /"?" confusion	Fraction of total in message index
mb5: the format for FOR EACH CASE OF is:	15/59
Total number of occurrences of this type:	15

Type 35: FOR EACH CASE OF using only a triple	Fraction of total in message index
mb5: the format for FOR EACH CASE OF is:	32/59
Total number of occurrences of this type:	32

Type 36: Attempt to get out of fancy features	Fraction of total in message index
mb3: I don't have a HELP option called	2/20
Total number of occurrences of this type:	2

Type 37: Attempt to use COPY as a HELP option	Fraction of total in message index
mb3: I don't have a HELP option called	2/20
Total number of occurrences of this type:	2

Type 38: Student in infinite loop	Fraction of total in message index
mb4: infinite loop (recursion) suspected	64/64
Total number of occurrences of this type:	64

BEST COPY AVAILABLE

Type 39: Attempt to CUPI a FUN EACH procedure before state change to FUN EACH mode — Fraction of total in message index

m71: can't CUPI 'FUN EACH' until you first use it yourself 2/2

Total number of occurrences of this type: 2

Type 40: Mixed syntax in subline prompt — Fraction of total in message index

m72: wrong subline prompt 1/1

Total number of occurrences of this type: 1

Type 41: Use of wild card in check, while in FUN EACH mode — Fraction of total in message index

m81: wild card here is taboo (type HELP for explanation) 21/21

Total number of occurrences of this type: 21

Type 42: User record not found in the data base — Fraction of total in message index

m86: Record for user "X" not found 3/3

Total number of occurrences of this type: 3

Type 43: Non-unique variable names in a global variable situation — Fraction of total in message index

m86: your * variable refers back to more than one 'FUN EACH' 4/4

Total number of occurrences of this type: 4

BEST COPY AVAILABLE

Type 44: FOR EACH or * used at top level	Fraction of total in message index
m15: the variable *X has no value at this point	1/59
m89: you may only use 'FOR EACH' within a procedure definition	1/2
Total number of occurrences of this type:	2

NOTE: Type 45 was discarded

Type 46: Use of controll flow statements where inappropriate	Fraction of total in message index
m37: Left Hand Quotes (") Assumed	39/155
m38: Right Hand Quotes (") Assumed	43/146
Total number of occurrences of this type:	82

Type 47: Delimiter problem	Fraction of total in message index
m37: Left Hand Quotes (") Assumed	11/155
m38: Right Hand Quotes (") Assumed	11/146
Total number of occurrences of this type:	22

Type 48: Confusion between /X/, X and *X as values in a PRINT statement	Fraction of total in message index
m37: Left Hand Quotes (") Assumed	18/155
m38: Right Hand Quotes (") Assumed	18/146
Total number of occurrences of this type:	36

BEST COPY AVAILABLE

Type 49: Multiple procedure calls on a single line	Fraction of total in message index
m37: Left Hand Quotes (`) Assumed	7/155
m38: Right Hand Quotes (') Assumed	7/148
Total number of occurrences of this type:	14

Type 50: Positive response to a confirmation statement	Fraction of total in message index
m113: Do you want to select any more options	129/573
Total number of occurrences of this type:	129

Type 51: Negitive response toa confirmation statement	Fraction of total in message index
m113: Do you want to select any more options	388/573
Total number of occurrences of this type:	388

Type 52: *A/A confusion in parameter list	Fraction of total in message index
m51: illegal characters	2/2
Total number of occurrences of this type:	2

Type 53: Wrong procedure call to a student defined procedure	Fraction of total in message index
m14: the format for user defined procedure is:	97/191
Total number of occurrences of this type:	97

BEST COPY AVAILABLE

Type 54: Student leaves machine and fails to respond to two previos warnings about auto log out	Fraction of total in message index
m3: you will be logged off if you don't type something at once	15/15
Total number of occurrences of this type:	15

Type 55: Attempted access built-in to SOLO procedures	Fraction of total in message index
m9: you can't LIST built-in SOLO procedures	1/2
Total number of occurrences of this type:	1

Type 56: Attempt to EDIT a built-in SOLO procedure	Fraction of total in message index
m9: you can't LIST built-in SOLO proceedures	1/2
Total number of occurrences of this type:	1

Type 57: Synonym corrector works	Fraction of total in message index
m27: the format for BYE is:	2/2
Total number of occurrences of this type:	2

Type 58: Attempt to NOTE an assertion that already exists in the data base	Fraction of total in message index
m28: OK... (I already knew [node relation node])	322/322
Total number of occurrences of this type:	322

BEST COPY AVAILABLE

Type 59: Attempt to FORGET an assertion that does not exist in the data base	Fraction of total in message index
m30: OK... (I didn't know (node relattion node))	116/116
Total number of occurrences of this type:	116

Type 60: Attempt to create a procedure which already exists	Fraction of total in message index
m31: I already have a procedure called	63/63
Total number of occurrences of this type:	63

Type 61: A positive response to redefining proceedures	Fraction of total in message index
m32: Would you like to redefine it now?	9/25
m33: Do you want to redefine it with	14/26
Total number of occurrences of this type:	23

Type 63: No response to a confirmation message	Fraction of total in message index
m33: Do you want to redefine it with	4/26
m101: Type option here:	21/50
Total number of occurrences of this type:	25

Type 64: Use of ' instead of " to delimit a string	Fraction of total in message index
m39: Please use " to end a quoted string and not '	11/11
Total number of occurrences of this type:	11

BEST COPY AVAILABLE

Type 65: student successfully concludes the creation of a proceedure (using TO)	Fraction of total in message index
m65: I now know how to	274/274
Total number of occurrences of this type:	274

Type 66: Student successfully concludes the editing of a proceedure using EDIT	Fraction of total in message index
m66: I have re-defined how to	339/339
Total number of occurrences of this type:	339

Type 67: Missing controll flow statement in conditional	Fraction of total in message index
m76: (Please specify EXIT, CONTINUE, or NEXT CASE after the ; below)	107/126
Total number of occurrences of this type:	107

NOTE: Types 68 & 69 have been discarded

Type 70: Student forgets line number while in edit mode	Fraction of total in message index
m12: you forgot the line number	116/184
Total number of occurrences of this type:	116

Type 71: Student attempts to use NEXT CASE as a controll flow in a CHECK statement that isn't within a FOR EACH	Fraction of total in message index
m76: (Please specify EXIT, CONTINUE, or NEXT CASE after the ; below)	4/126
Total number of occurrences of this type:	4

BEST COPY AVAILABLE

Type 72: Line number out of legal range	Fraction of total in message index
m12: you forgot the line number	27/184
Total number of occurrences of this type:	27

Type 73: Reference to a variable which hasn't been assigned a value yet	Fraction of total in message index
m13: the variable *X has no value at this point	56/59
Total number of occurrences of this type:	56

Type 74: Student attempts to create a procedure with the same name as a built in SOLO procedure	Fraction of total in message index
m11 there's already a built in SOLO procedure called	1/2
Total number of occurrences of this type:	1

Type 75: Addition of sublines (e.g. 2A) where one is not legal	Fraction of total in message index
m5: I don't have a procedure called (user defined)	16/369
m57: param1 param2 param3 (at most 3 parameters)	3/21
m100: This number is reserved for the spelling corrector	21/310
Total number of occurrences of this type:	40

Type 76: Attempts multiple nodes from a single node-relation pair	Fraction of total in message index
m102: Oops...A---A--->A already	163/163
Total number of occurrences of this type:	163

BEST COPY AVAILABLE

Type 77: O used instead of 0 or instead of 1 — Fraction of total in message index

m57: param1 param2 param3 (at most 3 parameters) 2/21

Total number of occurrences of this type: 2

Type 78: Type in student number after already logging in — Fraction of total in message index

m5: I don't have a procedure called (user defined) 11/309

Total number of occurrences of this type: 11

Type 79: Attempt to have an interactive program — Fraction of total in message index

m5: I don't have a procedure called (user defined) 2/309

Total number of occurrences of this type: 2

Type 80: User uses a synonym for a SOLO procedure for which no synonym exists in the spelling corrector — Fraction of total in message index

m5: I don't have a procedure called (user defined) 3/309

m100: this number is reserved for the spelling corrector 1/310

Total number of occurrences of this type: 4

Type 81: Error message while in "CAT" option — Fraction of total in message index

m5: I don't have a procedure called (user defined) 121/309

Total number of occurrences of this type: 121

BEST COPY AVAILABLE

Type 82: Format error in BRIEF mode	Fraction of total in message index
m45: Wrong Format	9/9
Total number of occurrences of this type:	9

Type 83: Spelling corrector makes correct change to input line	Fraction of total in message index
m100: this number is reserved for the spelling corrector	271/310
Total number of occurrences of this type:	271

Type 84: Spell corrector gives the wrong split to a user input line	Fraction of total in message index
m100: this number is reserved for the spelling corrector	8/310
Total number of occurrences of this type:	8

Type 85: Spelling checker asks for confirmation on a plausible (word +/- 2 chars) spelling change; Accepted by user	Fraction of total in message index
m91: When you typed 'X' did you mean Y?	130/369
Total number of occurrences of this type:	130

Type 86: Spelling checker asks for confirmation on a plausible (word +/- 2 chars) spelling change; Rejected by user	Fraction of total in message index
m91: When you typed 'X' did you mean Y?	49/369
Total number of occurrences of this type:	130

BEST COPY AVAILABLE

Type 87: Spelling checker does a silly number check	Fraction of total in message index
m91: When you typed 'X' did you mean Y?	95/369
Total number of occurrences of this type:	95

Type 88: Spelling checker tries a ridiculous possible match (word +/- > 2 chars)	Fraction of total in message index
m91: When you typed 'X' did you mean Y?	95/369
Total number of occurrences of this type:	95

BEST COPY AVAILABLE

Appendix C: Breakdown by Message Number

The following pages give the breakdown of the occurrences of each message into the different types that received that message. Included is a summary of how many occurrences are trappable or pre-emptable, how many need new or improved messages, and how many are handled sufficiently (ok) by the current implementation.

Appendix C: Index by Message Number

BEST COPY AVAILABLE

MESSAGE NUMBER 3

Message name: (Sorry... you will be logged off if you don't type something at once)

Summary statistics

Total number of occurrences: 15
Total number of proper occurrences of error message(ok): 15
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t54) Student leaves machine and fails to respond to two previous warnings about auto log out.
e.g. (8 minutes have elapsed. Please press the RETURN key). (16 minutes have elapsed! Are you still there? Please press return key) (sorry... you will be logged off if you don't type something at once)
number of occurrences: 15
syntactic category: none
conceptual category: none
status: ok
priority: na

MESSAGE NUMBER 4

Message name: you typed too much for me to deal with at one time.

Summary statistics

Total number of occurrences: 40
Total number of proper occurrences of error message(ok): 30
Number of trapable errors: 6
number of errors which need a new or improved error message: 4

Error Types

t3) The user adds an unwanted space to the node/relation or procedure name after showing that s/he understands the format
e.g. NOTE FIDO LIKE S BEER
NOTE ONE IS A NUMBER
number of occurrences: 6
syntactic category: spelling error
status: trapable
priority: low RTC: 2

t8) Errors due to type ahead problems
e.g. NOTE FIDO ISA DOG FORGET FIDO LIKES BEER
number of occurrences: 2
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

BEST COPY AVAILABLE

t11) Over legal limit for number of characters in a node or procedure name
e.g. NOTE RICHARD ISA WISHYWASHYONE
number of occurrences: 4
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high RTC: 4

MESSAGE NUMBER 5

Message name: I don't have a procedure called X

Summary statistics

Total number of occurrences: 369
Total number of proper occurrences of error message(ok): 110
number of trapable errors: 131
number of errors which need a new or improved error message: 128

Error Types

t28A No referent
e.g. LIST FOO
(there is no procedure called FOO)
number of occurrences: 95
syntactic category: none
conceptual category: possibly a forward declaration of an as-yet-undefined procedure
status: ok
priority: na

t26) Forgot to include an obvious SOLO procedure name
e.g. SOLO: 6---ismorethan--- 7
number of occurrences: 34
syntactic category: missing 'procedure name, arguments only
conceptual category: none
status: trapable
priority: medium RTC: 2

t78) Type in student number after already logging in
e.g. SOLO: BØ1191X
Oops... I don't have a procedure called BØ1191X
number of occurrences: 11
syntactic category: not a legal procedure name
conceptual category: either type-ahead or doesn't believe that s/he has successfully logged in.
status: trap
priority: medium RTC: 2

t29b) Attempt to use LIST on a node
e.g. LIST FIDO
number of occurrences: 8
syntactic category: mixed arg types
conceptual category: data/procedure confusion, concept of a general "show me" procedure
status: improve message
priority: medium

BEST COPY AVAILABLE

t30) Use Describe as an invocation function
e.g. DESCRIBE DEMO1
number of occurrences: 2
syntactic category: procedure arg in place of data arg
conceptual category: necessary invocation program
status: ok
priority: na RTC: 3

t80) User uses a synonym for a SOLO procedure for which no synonym exists in the spelling corrector
e.g. DEFINE SUBCHECK (instead of TO)
ADD FIDO ISA DOG (instead of NOTE)
number of occurrences: 3
syntactic category: illegal procedure name
conceptual category: use of synonym instead of legal procedure name
status: ok
priority: na RTC:

t18b) Mode confusion: student assumed s/he was in editor
e.g. SOLO: 10AA
number of occurrences: 10
syntactic category: none
conceptual category: mode error
status: improve message
priority: medium RTC: 3

t32) Spelling error
e.g. NPOTE 10 SUBTABLE 10RB
number of occurrences: 23
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t17) Wrong access to the help features
e.g. SELECT OPTION or TRACE SUBTRACT or SILENCE (All of these at top level)
number of occurrences: 16
syntactic category: wrong mode for input
conceptual category: help mode accessible directly from SOLO top level
status: trapable
priority: medium RTC: 3

t81) Error message while in "CAT" option
e.g. (procedures refer to other procedures which have yet to be defined)
Oops... I don't have a procedure called X
number of occurrences: 121
syntactic category: none
conceptual category: undefined procedures still referenced in other procedures
status: improve message
priority: low RTC: 1 EST: 00:00

BEST COPY AVAILABLE

t11) Over legal limit for number of characters in a node or procedure name
e.g. 2A: SETFLAGSINDATA
(at runtime will generate the error message as if it doesn't exist)
number of occurrences: 8
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high RTC: 4

t24) Editing node names
e.g. EDIT FIDO
number of occurrences: 3
syntactic category: giving data as an argument in place of a procedure
conceptual category: data/procedure confusion, argument requirement confusion
status: improve message
priority: medium RTC: 3

t27) Gave single number at top level
e.g. SOLO: 6 (there is a nasty BASIC error message in response to this)
number of occurrences: 7
syntactic category: no procedure name given or defined for the number
conceptual category: perhaps a hanging input mode confusion
status: improve message
priority: high RTC 2

t4) Flow of control statement taken as an argument
e.g. ....2A If Present continue; continue
number of occurrences: 13
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium RTC: 3

t79) Attempt to have an interactive program
e.g. SOLO: SUBTRACT
OK... THE ANSWER IS 109
DO YOU WANT TO TRY ANOTHER PROBLEM?
SOLO: YES
Oops... I don't have a procedure called YES
number of occurrences: 2
syntactic category: thinks I/O is possible with running program
status: ok
priority: na

BEST COPY AVAILABLE

t75) Addition of sublines (e.g. 2A) where one is not legal
e.g. ...:2A PRINT "HI"
[line 2 doesn't require a substep]
(2A = 2 A) Oops.I don't have a procedure called A (at run time)
number of occurrences: 16
syntactic category: subline not legal
conceptual category: subline construct legal in all places or made an error editing
status: trap
priority: High RTC: 4

MESSAGE NUMBER 9

Message name: you can't LIST built-in SOLO Procedures (like NOTE, etc.)

Summary statistics

Total number of occurrences: 2
Total number of proper occurrences of error message(ok): 1
number of trapable errors: 0
number of errors which need a new or improved error message: 1

Error Types

t55) Attempted access to "built-in" SOLO procedure
e.g. LIST HELP
number of occurrences: 1
syntactic category: none
conceptual category: system procedure changes allowed or simple type ahead problem
status: improve message
priority: high RTC: 3

t56) Attempt to EDIT a built in SOLO procedure
e.g. EDIT NOTE
number of occurrences: 1
syntactic category: none
conceptual category: system procedure changes allowed
status: trap
priority: na RTC: 3

MESSAGE NUMBER 11

Message name: There's already a built-in SOLO procedure called
Total number of occurrences: 2

Error Types

t30) Use "TO" as an invocation
e.g. TO EDIT PRAISE
number of occurrences: 1
syntactic category: procedure arg in place of data arg
conceptual category: necessary invocation program
status: ok
priority: na RTC: 3

BEST COPY AVAILABLE

t74) Student attempts to create a procedure with the same name as a built in SOLO procedure
e.g. TO LIST
number of occurrences: 1
syntactic category: none
conceptual category: reserved name violation
status: improve
priority: low RTC: 3

MESSAGE NUMBER 12

Message name: You forget the line number

Summary statistics

Total number of occurrences: 184
Total number of proper occurrences or error message(ok): 116
number of trapable errors: 41
number of errors which need a new or improved error message: 27

Error Types

t70) Student forgets line number while in edit mode
e.g. TO PRAISE /X/
PRINT "I THINK THAT" /X/
Oops... you forgot the line number
number of occurrences: 116
syntactic category: missing line number
conceptual category: none
status: ok
priority: na

t18a) Mode error, probably due to type ahead
e.g. ...: 9 PRINT "ALMOST THERE"
...: LIST FOO
Oops you forgot the line number
number of occurrences: 39
syntactic category: none
conceptual category: mode error
status: improve message
priority: medium RTC: 3

t32) Spelling error
e.g. ...: DOOLNEE
(wanted to type DONE)
number of occurrences: 2
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t72) Line number out of range
e.g. ...: Ø Print "THIS IS THE FIRST LINE"
or
...: 15Ø Print "THIS IS FUN"
Oops... You forgot the line number
number of occurrences: 27
syntactic category: out of legal bounds of a variable
conceptual category: none
status: new message
priority: high RTC:3

BEST COPY AVAILABLE

MESSAGE NUMBER 13

Message name: The variable *X has no value (no 'contents') at this point.

Summary statistics

Total number of occurrences: 59
Total number of proper occurrences of error message(ok): 56
number of trapable errros: 2
number of errors which need a new or improved error message: 1

Error Types

t73) Reference to a variable which hasn't been assigned a value yet
SOLO: TO FOO
e.g....: 1 PRINT "THE ANSWER IS" *A
number of occurrences: 2
syntactic category: none
conceptual category: mixing up variable and parameter concepts, or scope of variable confusion
status: trap
priority: high RTC: 4

t18b) Mode confusion: student assumed s/he was in editor
e.g. SOLO: 3 Print *
number of occurrences: 2
syntactic category: none
conceptual category: mode error
status: improve message
priority: medium RTC: 3

t44) FOR EACH or * variable called at top level
e.g. Print *
number of occurrences: 1
syntactic category: wrong mode for the construct
conceptual category: mode confusion, or misunderstanding constraints of construct
status: ok
priority: na

MESSAGE NUMBER 14

Message name: The format for is: /parameter1/parameter2/parameter3/

Summary statistics

Total number of occurrences: 191
Total number of proper occurrences of error message(ok): 169
number of trapable errors: 14
number of errors which need a new or improved error message: 8

Error Types

BEST COPY AVAILABLE

t53) Wrong procedure call to a student defined procedure.
e.g. FOO /ME/ /YOU/
(FOO only takes 1 arg in this case)
number of occurrences: 97
syntactic category: wrong number of args to procedure
conceptual category: none
status: improve message
priority: High RTC: 2

t12) Spurious control characters in node or relation names
e.g. FOO ME^G You
number of occurrences: 5
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority high RTC: 2

t8) Errors due to type ahead problems
e.g. Addup 3 4 list Addup
number of occurrences: 4
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

t6) Too few args because of missing space between
e.g. TOPNUM 12
number of occurrences: 30
syntactic category: spelling error
conceptual category: none
status: trapable
priority: medium RTC: 2

t21) Use more than three args to a user defined procedure
e.g. SUBTRACT 1 2 3 4
number of occurrences: 12
syntactic category: too many args to a procedure
conceptual category: thinks that an unlimited number of args are possible
status: ok
priority: na

t5) Too few arguments given to a procedure
e.g. TOPNUM 1 (should get 2 numbers),
number of occurrences: 19
syntactic category: not enough args
conceptual category: none
status: improve message
priority: medium RTC: 2

t2) Spell corrector fault giving more than allowed number of arguments
e.g. numberbase_more 3 4
(SOLO corrects to be old procedure name numberbase + three args)
number of occurrences: 5
syntactic category: not enough args
conceptual category: none
status: improve message
priority: medium RTC: 2

BEST COPY AVAILABLE

t4) Flow of control statement taken as an argument
e.g. ...: if present: TAKEAWAY /Z/continue
number of occurrences: 4
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium RTC: 3

t7) No arguments given
e.g. FOO
number of occurrences: 7
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t11) Over legal limit for number of characters in a node or procedure name
e.g. FOO QUICKGETAWAYS BANDITS
number of occurrences: 8
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high RTC: 4

MESSAGE NUMBER 15, 16, 17

Message name: The format for Print is:
PRINT "quoted-string"

Summary statistics

Total number of occurrences: 4
Total number of proper occurrences of error message(ok): 3
number of trapable errors: 1
number of errors which need a new or improved error message: 0

Error Types

t7) No arguments given
e.g. 3A If present: Print; Exit
number of occurrences: 3
syntactic category: missing args
conceptual category: none
status: ok
priority: na

t13) Steamroller effect by the synonym and spelling corrector
e.g. 80 OUTPUT
(OUTPUT=PRINT)
Oops... the format for print is:
number of occurrences: 1
syntactic category: none
conceptual category:
status: trapable
priority: high RTC: 5

BEST COPY AVAILABLE

MESSAGE NUMBER 18

Message name: Oops... the format for NOTE is
NOTE node---relation--->node

Summary statistics

Total number of occurrences: 156
Total number of proper occurrences of error message(ok): 68
number of trapable errors: 45
number of errors which need a new or improved error message: 48

Error Types

t1) Too many arguments to procedure wording
e.g. NOTE FIDO LIKES WEST HAM UNITED
number of occurrences: 19
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low

t2) Spell corrector fault giving more than three args
e.g. NOTEW FIDO LIKES BEER
(The spelling corrector turns this into: NOTE W FIDO LIKES and in turn produces an error message for the wrong form
number of occurrences: 2
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t3) The user adds an unwanted space to the node/relation name after showing that s/he understands the format
e.g. NOTE FIDO LIKE S BEER
NOTE ONE IS A NUMBER
number of occurrences: 19
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t4) Flow of control statement taken as an argument in the note
e.g. if present: NOTE FIDO ISA DOG CONTINUE
number of occurrences: 4
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium

t5) Too few arguments given to note
e.g. NOTE JANE LIKES
number of occurrences: 35
syntactic category: not enough args
conceptual category: none
status: ok
priority: na

BEST COPY AVAILABLE

(6) Too few args because of missing space between node and between relation name
e.g. NOTE FIDO LIKESBEER
number of occurrences: 22
syntactic category: spelling error
conceptual category: none
status: ok
priority: na

(7) No args given to NOTE
e.g. NOTE
number of occurrences: 6
syntactic category: args problem
conceptual category: none
status: ok
priority: na

(8) Errors due to type ahead problems
e.g. NOTE FIDO ISA DOG FORGET FIDO LIKES BEER
number of occurrences: 6
syntactic category: none
conceptual category: none
status: ok
priority: na

(9) Used "?" in a node in the NOTE procedure
e.g. NOTE FIDO ISA ?
number of occurrences: 17
syntactic category: formal/actual variable name reversal
conceptual category: variable name confusion
status: improved message
priority: medium

(10) Had illegal characters in node or relation names
e.g. NOTE FID;O LIKES BEER
number of occurrences: 4
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high

(11) Over legal limit for number of characters in a node name
e.g. NOTE RICHARD ISA WISHYWASHYONE
number of occurrences: 12
syntactic category: character limit overflow
conceptual category: none
status: differrent error message
priority: high

(12) Spurious control characters in node or relation names
e.g. NOTE FIDO IS↑A DOG
number of occurrences: 15
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

t13) Steamroller effect by the synonym and spelling corrector
e.g NATE
(SOLO makes this into NOTE and gives the error message)
number of occurrences: 1
syntactic category: none
conceptual category: none
status: trapable
priority: high

MESSAGE NUMBER 19

Message name: Oops... the format for DESCRIBE is DESCRIBE node

---

Summary statistics

Total number of occurrences: 24
Total number of proper occurences of error message(ok): 8
number of trapable errors: 4
number of errors which need a new or improved error message: 12

Error types

t1) Too many arguments to DESCRIBE
e.g. DESCRIBE FIDO MARY NIXON
number of occurrences: 12
syntactic category: too many args
conceptual category: procedures can take multiple args
status: improve message
priority: low

t2) Spell corrector fault giving too many args
e.g. DESCRIBEW FIDO
(The spelling corrector turns this into: DESCRIBE W FIDO and in turn produces an error message for the wrong format)
number of occurrences: 2
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t3) The user adds an unwanted space to the node/relation name after showing that s/he understands the format
e.g. DESCRIBE ADDTABLE 1
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t4) No args given to DESCRIBE
e.g. DESCRIBE
number of occurrences: 8
syntactic category: args problem
conceptual category: none
status: ok
priority: na

BEST COPY AVAILABLE

t5) Errors due to type ahead problems
e.g. DESCRIBE FIDO FORGET FIDO LIKES BEER
number of occurrences: 0
syntactic category: none
conceptual category: none
status: ok
priority: na

t6) Had illegal characters in node or relation names
e.g. DESCRIBE FIDO/
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high

MESSAGE NUMBER 20

Message name: Oops... the format for FORGET is
FORGET node---relation--- node

Summary statistics

Total number of occurrences: 42
Total number of proper occurrences of error message(ok): 28
number of trapable errors: 11
number of errors which need a new or improved error message: 3

Error Types

t1) Too many arguments to FORGET
e.g. FORGET FIDO LIKES WEST HAM UNITED
number of occurrences: 1
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low

t2) Spell corrector fault giving more than three args
e.g. FORGETW FIDO LIKES BEER
(The spelling corrector turns this into: FORGET W FIDO and in turn produces an error message for the wrong format)
number of occurrences: 4
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t3) The user adds an unwanted space to the node/relation name after showing that s/he understands the format
e.g. FORGET FIDO LIKES BEE R
FORGET ONE IS A NUMBER
number of occurrences: 2
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

BEST COPY AVAILABLE

t4) Flow of control statement taken as an argument
e.g. if present: FORGET FIDO ISA DOG CONTINUE
number of occurrences: 4
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium

t5) Too few arguments given
e.g. FORGET JANE LIKES
number of occurrences: 20
syntactic category: not enough args
conceptual category: none
status: ok
priority: na

t6) Too few args because of missing space between node and relation node
e.g. FORGET FIDO LIKESBEER
number of occurrences: 5
syntactic category: spelling error
conceptual category: none
status: ok
priority: na

t7) No args given to FORGET
e.g. FORGET
number of occurrences: 3
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t8) Errors due to type ahead problems
e.g. FORGET FIDO ISA DOG FORGET FIDO LIKES BEER
number of occurrences: 0
syntactic category: none
conceptual category: none
status: ok
priority: na

t9) Used "?" in a node in the FORGET procedure
e.g. FORGET FIDO ISA ?
number of occurrences: 1
syntactic category: formal/actual variable name reversal
conceptual category: variable name confusion
status: improved message
priority: medium

t10) Had illegal characters in node or relation names
e.g. FORGET FIDO; LIKES BEER
number of occurrences: 0
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

(11) Over legal limit for number of characters in a node name
e.g. FORGET RICHARD ISA WISHYWASHYONE
number of occurrences: 0
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high

(12) Spurious control characters in node or relation names
e.g. FORGET FIDO IS A DOG
number of occurrences: 0
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

(13) Steamroller effect by the synonym and spelling corrector
e.g. DELETE
(SOLO makes this student defined procedure into FORGET and gives the error message)
number of occurrences: 1
syntactic category: none
conceptual category: none
status: trapable
priority: high

MESSAGE NUMBER 23

Message name: the format for EDIT is
EDIT procedure

Summary statistics

Total number of occurrences: 22
Total number of proper occurrences of error message(ok): 8
number of trapable errors: 12
number of errors which need a new or improved error message: 2

Error Type:

(1) Too many arguments
e.g. EDIT PRAISE FOO
number of occurrences: 2
syntactic category: too many args
conceptual category: multiple args to a procedure assumed
status: improve message
priority: low

(7) No arguments given
e.g. EDIT
number of occurrences: 5
syntactic category: args problem
conce

BEST COPY AVAILABLE

t8) Errors due to type ahead problems
e.g. EDIT PRAISE 3 PRINT "HELLO"
number of occurrences: 3
syntactic category: none
conceptual category: none
status: ok
priority: na

t12) Spurious control characters in node or relation names
e.g. EDIT PR↑AISE
number of occurrences: 1
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

t23a) Gave procedure name with TO attached to it or the parameters included.
e.g. EDIT TO FOO
number of occurrences: 8
syntactic category: included procedure definition term in procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high

t29b) Attempt to use EDIT on a node
e.g. EDIT FIDO
number of occurrences: 3
syntactic category: mixed arg types
conceptual category: data/procedure confusion, wanted to NOTE or FORGET something
status: trapable
priority: medium

MESSAGE NUMBER 22

Message name: the format for LIST is LIST procedure

Summary statistics

Total number of occurrences: 24
Total number of proper occurrences of error message(ok): 3
number of trapable errors: 17
number of errors which need a new or improved error message: 4

Error Types

t1) Too many arguments
e.g. LIST FOO SUSS PRAISE
number of occurrences: 2
syntactic category: too many args
conceptual category: procedures can take multiple args
status: improve message
priority: low

BEST COPY AVAILABLE

t3) The user adds an unwanted space to the procedure name after showing that s/he understands the format
e.g. LIST SUS S
number of occurrences: 4
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t7) No arguments given
e.g. LIST
number of occurrences: 1
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t8) Errors due to type ahead problems
e.g. (simple type ahead error)
number of occurrences: 2
syntactic category: none
conceptual category: none
status: ok
priority: na

t10) Had illegal characters in procedure name
e.g. LIST PRAISE/3
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high

t11) Over legal limit for number of characters in a node name
e.g. LIST SUSSTHESITUATION
number of occurrences: 2
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high

t23a) Gave procedure name with TO attached to it
e.g. LIST TO FOO
number of occurrences: 5
syntactic category: included procedure definition term in procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high

t23b) Gave procedure name with args included to a procedure
e.g. LIST FOO /X/ /Y/
number of occurrences: 7
syntactic category: included args as part of procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high

BEST COPY AVAILABLE

MESSAGE NUMBER 24

Message name: the format for TO is TO procedure (/parameter1/
/parameter2/ /parameter 3/)

Summary statistics

Total number of occurrences: 33
Total number of proper occurrences of error message(ok): 5
number of trapable errors: 23
number of errors which need a new or improved error message: 5

Error Types

t3) The user adds an unwanted space to the node/relation name
after showing that s/he understands the format
e.g. TO LOOK UP /X/
number of occurrences: 6
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t8) Errors due to type ahead problems
e.g TO FOO /X/ 1 CHECK /X/ wants ?
number of occurrences: 3
syntactic category: none
conceptual category: none
status: ok
priority: na

t11) Over legal limit for number of characters in a procedure name
e.g. TO DOFINALNUMBER /X/
number of occurrences: 5
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high

t12) Spurious control characters in procedure names
e.g. TO PRAISE /X/
number of occurrences: 6
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

t13) Steamroller effect by the synonym and spelling corrector
e.g. TOTAL 1 2
number of occurrences: 8
(SOLO makes this into TO TAL 1 2 and gives the error
message)
syntactic category: none
conceptual category: none
priority: high

BEST COPY AVAILABLE

t21) Use more than three args to a user defined procedure
e.g. TO LOOKUP /W/ /X/ /Y/ /Z/
number of occurrences: 2
syntactic category: two many args to a procedure
conceptual category: thinks that an unlimited number of args are passable
status: ok
priority: na

t22) Forgot "/" in the procedure line during definition
e.g. TO LOOKUP X/
number of occurrences: 3
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: high

t19) Attempt to apply a procedure to a procedure
e.g. FORGET TO FOO or FORGET FOO
number of occurrences: 1
syntactic category: violation of defined syntax
conceptual category: need to 'obliterate' unwanted procedure
status: improved message
priority: medium RTC: 2

MESSAGE NUMBER 25

Message name: the format for CHECK is
CHECK node---relation--- ?
CHECK node---relation--- node

Summary statistics

Total number of occurrences: 44
Total number of proper occurrences of error message(ok): 16
number of trapable errors: 23
number of errors which need a new or improved error message: 5

Error Types

t1) Too many arguments
e.g. CHECK FIDO ISA BIG DOG
number of occurrences: 3
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low

t3) The user adds an unwanted space to the node/relation name after showing that s/he understands the format
e.g. CHECK FIDO IS A DOG
number of occurrences: 4
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

BEST COPY AVAILABLE

t4) Flow of control statement taken as an argument
e.g. if present: CHECK FIDO ISA DOG CONTINUE
number of occurrences: 1
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium

t5) Too few arguments given
e.g. CHECK FIDO ISA
number of occurrences: 10
syntactic category: not enough args
conceptual category: none
status: ok
priority: na

t6) Too few args because of missing space between node and relation name
e.g. CHECK FIDO LIKESBEER
number of occurrences: 3
syntactic category: spelling error
conceptual category: none
status: trapable
priority: na

t7) No arguments given
e.g. CHECK
number of occurrences: 3
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t11) Over legal limit for number of characters in a node name
e.g. CHECK RICHARD ISA WISHYWASHYONE
number of occurrences: 2
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high

t12) Spurious control characters in node or relation names
e.g. CHECK FIDO ISA DOG
number of occurrences: 6
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

t13) Steamroller effect by the synonym and spelling corrector
number of occurrences 4
(SOLO makes this into CHECK /X/ /Y/ and gives the error message)
syntactic category: none
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

t14) Over extended use of wild card
e.g. CHECK ?---LIKES--- BEER
number of occurrences: 1
syntactic category: variable syntax violation
conceptual category: over generalization of variable placement
status: trapable
priority: high

t18a) Mode confusion; thought in top level
e.g. ...: EDIT
number of occurrences: 3
syntactic category: none
conceptual category: mode error
status: trapable
priority: medium

t20) Attempted concatenation of strings
e.g. CHECK X "A"/ANY/ ?
number of occurrences: 4
syntactic category: putting strings into node spaces
conceptual category: thinks that concatenation is possible
status: trapable
priority: low

MESSAGE NUMBER 27

Message name: The format for BYE is:
BYE

Summary statistics

Total number of occurrences: 2
Total number of proper occurrences of error message(ok): 2
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t57) Synonym corrector works
e.g. SOLO: END RUN
Oops the format for BYE is:
BYE
number of occurrences: 2
syntactic category: wrong command word
conceptual category: none
status: ok
priority: na

BEST COPY AVAILABLE

MESSAGE NUMBER 28

Message name: O.K. ... (I already know A---A--- A)

Summary statistics

Total number of occurrences: 322
Total number of proper occurrences of error message(ok): 322
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t58) Notes an assertion which already exists in the data base
e.g.NOTE FIDO---ISA--- DOG
OK (I already knew FIDO ISA DOG)
number of occurrences: 322
syntactic category: none
conceptual category: over-writing existing assertions is legal
status: improved message
priority: high RTC: 2

MESSAGE NUMBER 29

Message name: I have no description of X

Summary statistics

Total number of occurrences: 148
Total number of proper occurrences of error message(ok): 123
number of trapable errors: 25
number of errors which need a new or improved error message: 0

Error Types

t3) The user adds an unwanted space to the node/relation name after showing that s/he understands the format
e.g. DESCRIBE FID O
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: low

t28a) No referant
e.g. DESCRIBE BASEBALL
(there is no node in the data base called BASEBALL)
number of occurrences: 106
syntactic category: none
conceptual category: possibly a missing desired effect or unexpected side effect
status: ok
priority: na

BEST COPY AVAILABLE

t28b) Describe a relation name with no referent
e.g. DESCRIBE LIKES
(there is no description of LIKES in the data base)
number of occurrences: 9
syntactic category: none
conceptual category: node/relation confusion
status: ok
priority: na

t29a) Attempt to use DESCRIBE on a procedure
e.g. DESCRIBE FOO
number of occurrences: 24
syntactic category: mixed args types
conceptual category: data/procedure confusion, concept of a general "show me" procedure
status: trapable
priority: medium

t30) Use describe as an invocation function
e.g. DESCRIBE DEMO1
number of occurrences: 6
syntactic category: procedure arg in place of data arg
conceptual category: necessary invocation program
status: ok
priority: na

t31) Attribution of too much knowledge to a function
e.g. DESCRIBE SOLO or DESCRIBE NUMBER
number of occurrences: 2
syntactic category: improper arg type
conceptual category: User thinks systems has human understanding capabilities
status: ok
priority: na

MESSAGE NUMBER 30

Message name: O.K.... (I didn't know B---B--- B

Summary statistics

Total number of occurrences: 116
Total number of proper occurrences of error message(ok): 116
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t59) Attempt to FORGET an assertion that does not exist in the data base
e.g. SOLO: FORGET FIDO ISA PIG
ok... (I didn't know FIDO ISA PIG)
number of occurrences: 116
syntactic category: none
conceptual category: Assumption that an assertion exists in the data base when it actually doesn't
status: Improved message
priority: high RTC: 2

BEST COPY AVAILABLE

MESSAGE NUMBER 31

Message name: I already have a procedure called X

Summary statistics

Total number of occurrences: 63
Total number of proper occurrences of error message(ok): 63
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t60) Attempt to create a procedure which already exists
e.g. SOLO: TO PRAISE /X/
Oops... I already have a procedure called 'PRAISE' /X/
number of occurrences: 63
syntactic category: possibly wrong call to the editor
conceptual category: creating and editing a file are synonymons
status: ok
priority: na

MESSAGE NUMBER 32 & 34

Message name: I already have a procedure called X
would you like to redefine it now? (Yes or No):

Summary statistics

Total number of occurrences: 25
Total number of proper occurrences of error message(ok): 2
number of trapable errors: 0
number of errors which need a new or improved error message: 23

Error Types

t8) Errors due to type ahead problems
e.g. would you like to redefine it now? (Yes or No): CHECK /X/
is BALDING
number of occurrences: 2
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

t61) Possible confirmation to redefining procedures
e.g. Would you like to redefine it now? (Yes or No): Y
Go ahead but remember that any procedures which use the old 'PRAISE' 'X' will have to be redefined.
number of occurrences: 9
syntactic category: none
conceptual category: none
status: improve message
priority: high

BEST COPY AVAILABLE

t62) Negative confirmation to redefining procedures
e.g. would you like to redefine it now? (Yes or No): N
Do you want to redefine it now? (Yes or No): NO
number of occurrences: 14
syntactic category: none
conceptual category: possible misunderstanding of message
status: improve message
priority: high RTC: 3

MESSAGE NUMBER 33

Message name: Do you want to redefine it with NEW parameters (Yes or No):

Summary statistics

Total number of occurrences: 26
Total number of proper occurrences of error message(ok): 0
number of trapable errors: 0
number of errors which need a new or improved error message:20

Error Type

t8) Errors due to type ahead problems. Do you want to redefine it now? (Yes or No): TO FOO /X/
number of occurrences: 2
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

t61) A positive response to redefining procedures
e.g. Do you want to redefine it with NEW parameter(s) (Yes or No): YES
number of occurrences: 14
syntactic category: none
conceptual category: may think this is the same as editing
status: improve message
priority: high RTC: 3

BEST COPY AVAILABLE

t62) Negative confirmation to redefining procedure
e.g. Do you want to redefine it with NEW parameter(s)?
(Yes or No): NO
number of occurrences: 5
syntactic category: none
conceptual category: possible misunderstanding of message
status: improve message
priority: high RTC: 3

t63) No response to a confirmation message
e.g. Do you want to redefine it now? (Yes or No):
Oops I already have a procedure called 'Praise' /X/
Do you want to redefine it now? (Yes or No):
number of occurrences: 4
syntactic category: no arg given
conceptual category: not understanding error
status: improve message
priority: high RTC: 3

MESSAGE NUMBER 36

Message name: (The line number was not necessary)...

Summary statistics

Total number of occurrences: 28
Total number of proper occurrences of error message(ok): 9
number of trapable errors: 16
number of errors which need a new or improved error message: 3

Error Types

t8) Errors due to type ahead problems
e.g. (the user's edit command is not accepted and s/he types ahead doing edit commands)
number of occurrences: 9
syntactic category: none
conceptual category: none
status: ok
priority: na

t18a) Mode confusion: thought in editor
e.g. SOLO: 2 NOTE X Y Z
number of occurrences: 8
syntactic category: none
conceptual category: mode error
status: trapable
priority: medium

t26) Forgot procedure name
e.g. SOLO: 6---ismorethan--- 7
number of occurrences: 8
syntactic category: missing procedure name, arguments only
conceptual category: none
status: trapable
priority: medium

BEST COPY AVAILABLE

t27) Gave single number at top level
e.g. SOLO: 6
number of occurrences: 3
syntactic category: no procedure name given or defined for the number
conceptual category: perhaps a hanging input mode confusion
status: improved message
priority: high

MESSAGE NUMBER 37

Message name: Left Hand Quotes (") Assumed

Summary statistics

Total number of occurrences: 155
Total number of proper occurrences of error message(ok): 64
number of trapable errors: 55
number of errors which need a new or improved error message: 36

Error Types

t4) Flow of control statement taken as an argument due to ignoring the first flow of control statement
e.g. if present: PRINT "BLAH" CONTINUE:CONTINUE
(SOLO puts "s around the first CONTINUE)
number of occurrences: 4
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium

17) Wrong access to the help features
e.g. PRINT OPTIONS
number of occurrences: 1
syntactic category: wrong mode for input
conceptual category: help mode accessible directly from SOLO top level
status: trapable
priority: medium

t25a) Forgot "which SOLO then correctly added
e.g. PRINT FIDO MUST BE A DOG"
number of occurrences: 64
syntactic category: missing delimiter
conceptual category: none
status: ok
priority: na

t12) Spelling error: used ":" instead of ";" as delimiter
e.g. ...if absent: PRINT "BAIL OUT": EXIT
(SOLO re-prompts for flow of control statement & quotes
EXIT)
number of occurrences: 11
syntactic category: mispell
conceptual category: none
status: trapable
priority: medium

BEST COPY AVAILABLE

t46) Use of control flow statements where inappropriate
e.g. ...: 1 PRINT "BLAH": CONTINUE
(CONTINUE is inappropriate here, and SOLO attempts to put it into the PRINT statement)
number of occurrences: 39
syntactic category: inappropriate use of control flow statement
conceptual category: control flow specification possible or necessary at any point in the procedure
status: trapable
priority: medium

t47) Delimiter problem
e.g. If present: PRINT "OKAY";/X/;CONTINUE
number of occurrences: 11
syntactic category: wrong delimiter used
conceptual category: none
status: improved message
priority: medium

t48) Confusion between /X/, X and *X as values in a PRINT statement
e.g. PRINT H X (should have been /H/ /X/)
(SOLO also throws away "/"s without mention if there is no parameter referent)
number of occurrences: 18
syntactic category: variable name/value/parameter denotation problem
conceptual category: variable name/value/parameter distinction
status: improved message
priority: medium

t49) Multiple procedure calls on a single line
e.g. .... if present: PRINT "GOT IT"; FOO; EXIT
number of occurrences: 7
syntactic category: illegal format
conceptual category: confusion about procedure call constraints
status: improved message
priority: medium

MESSAGE NUMBER 38

Message name: Right Hand Quotes (") Assumed

Summary statistics

Total number of occurrences: 146
Total number of proper occurrences of error message(ok): 51
number of trapable errors: 59
number of errors which need a new or improved error message: 36.

Error Types

BEST COPY AVAILABLE

t4) flow of control statement taken as an argument due to ignoring the first flow of control statement
e.g. if present: PRINT "BLAH"CONTINUE;CONTINUE
(SOLO puts "s around the first CONTINUE)
number of occurrences: 4
syntactic category: missing delimiter
conceptual category: none
status: trapable
priority: medium

t17) Wrong access to the help features
e.g. PRINT OPTIONS
number of occurrences: 1
syntactic category: wrong mode for input
conceptual category: help mode accessible directly from SOLO top level
status: trapable
priority: medium

t25a) Forgot "which SOLO then correctly added
e.g. PRINT "FIDO MUST BE A DOG
number of occurrences: 51
syntactic category: missing delimiter
conceptual category: none
status: ok
priority: na

t32) Spelling error; used ":" instead of ";" as delimiter
e.g. ...if absent: PRINT "BAIL OUT": EXIT
(SOLO re-prompts for flow of control statement & quotes EXIT)
number of occurrences: 11
syntactic category: mispell
conceptual category: none
status: trapable
prioirty: medium

t46) Use of control flow statements where inappropriate
e.g. ...: 1 PRINT "BLAH";CONTINUE
(CONTINUE is inappropriate here, and SOLO attempts to puts it into the PRINT statement)
number of occurrences: 43
syntactic category: inappropriate use of control flow statement
conceptual category: control flow specification possible or necessary at any point in the procedure
status: trapable
priority: medium

t47) Delimiter problem
e.g. If present: PRINT "OKAY";/X/;CONTINUE
number of occurrences: 11
syntactic category: wrong delimiter used
conceptual category: none
status: improved message
priority: medium

BEST COPY AVAILABLE

t48) Confusion between /X/, X and *X as values in a PRINT statement
e.g. PRINT H X (should have been /H/ /X/)
(SOLO also throws away "/"s without mention if there is no parameter referent)
number of occurrences: 18
syntactic category: variable name/value/parameter denotation problem
conceptual category: variable name/value/parameter distinction
status: improved message
priority: medium

t49) Multiple procedure calls on a single line
e.g. ....if present: PRINT "GOT IT"; FOO;EXIT
number of occurrences: 7
syntactic category: illegal format
conceptual category: confusion about procedure call constraints
status: improved message
priority: medium

MESSAGE NUMBER 39 & 40

Message name: Please use " to end a quoted string and not '
(Also check use of shift key)

Summary statistics

Total number of occurrences: 11
Total number of proper occurrences of error message(ok): 11
number of trapable errors: 0
number of errors which need a new or improved error message:0

Error Types

t64) Use of ' instead of " to delimit a string
e.g. Print 'HELLO WORLD"
(' = "7
Please use " to end a quoted-string, and not '
Also check use of SHIFT key
number of occurrences: 11
syntactic category: wrong delimiter
conceptual category: none
status: ok
priority: na

MESSAGE NUMBER 43

Message name: Wrong Format

Summary statistics

Total number of occurrences: 9
Total number of proper occurrences of error message(ok): 9
number of trapable errors: 0
number of errors which need a new or improved error message: 0

BEST COPY AVAILABLE

Error Types

t1) Too many arguments to a procedure
e.g. NOTE HENS---LAY---EGGS--- 0
number of occurrences: 7
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low RTC: 2

t7) No arguments given
e.g. 3 PRINT
number of occurrences: 2
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t82) Format error in BRIEF mode
e.g. NOTE FIDO I
number of occurrences: 9
syntactic category: format error, not enough Args
conceptual category: none
status: ok
priority: na

MESSAGE NUMBER 44

Message name: HELP not currently available for the above

Summary statistics

Total number of occurrences: 14
Total number of proper occurrences of error message(ok): 0
number of trapable errors: 14
number of errors which need a new or improved error message: 0

Error types

t13) Steamroller effect by the synonym and spelling corrector
e.g. HEL LO
number of occurrences: 1
(SOLO makes this into HELP and gives the error message)
syntactic category: none
conceptual category: none
status: trapable
priority: high

t17) Wrong access to help features
e.g. HELP CAT
number of occurrences: 4
syntactic category: wrong mode for input
conceptual category: help mode accessible directly from SOLO top level
status: trapable
priority: medium

BEST COPY AVAILABLE

t32) Spelling error
e.g. HELP OPTION
(SOLO requires HELP OPTIONS)
number of occurrences: 9
syntactic category: mispell
conceptual category: none
status: trapable
priority: medium

MESSAGE NUMBER 51

Message name: Illegal parameter name

Summary statistics

Total number of occurrences: 2
Total number of proper occurrences of error message(ok): 2

number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t52) *A/A confusion in parameter list
e.g. TO FOO /*A/
(SOLO corrects this to be TO FOO /A/)
number of occurrences: 2
syntactic category: parameter syntax violation
conceptual category: variable/parameter confusion
status: ok
priority: ok

MESSAGE NUMBER 53

Message name: invalid procedure name

Summary statistics

Total number of occurrences: 1
Total number of proper occurrences of error message(ok): 1
number of trapable errors: 0
number of errors which need a new of improved error message: 0

Error Types

t10) Illegal characters in procedure name
e.g. SOLO: @MATCH@
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high RTC: 2

BEST COPY AVAILABLE

```
MESSAGE NUMBER 54

Message name: Too many lines, only 10 allowed
-------------------------------------------------------------------------
Summary statistics

Total number of occurrences: 12
        Total number of proper occurrences of error message(ok): 12
        number of trapable errors: 0
        number of errors which need a new or improved error message: 0
-------------------------------------------------------------------------
Error Types

     t33) Too many lines in procedure
          e.g. (student uses more than 10 numbered lines)
          number of occurrences: 12
                syntactic category: none
                conceptual category: overflowing a preset limit
                status: ok
                priority: na
MESSAGE NUMBER 55

Message name:   Oops... the format for FOR is
                FOR EACH CASE OF node---relation--- ?x
-------------------------------------------------------------------------
Summary statistics

Total number of occurrences: 59
          Total number of proper occurrences of error message(ok): 5
          number of trapable errors: 39
          number of errors which need a new or improved error message:15
-------------------------------------------------------------------------
Error Types

     t2) Spell corrector fault gives more than allowed number of args
          e.g. FORE EACH CASE OF X Y ?
               (The spelling corrector turns this into: FOR E EACH CASE
               OF and in turn produces an error message for the wrong
               format)
          number of occurrences: 7
               syntactic category: spelling error
               conceptual category: none
               status: trapable
               priority: low

     t7) No arguments given
          e.g. FOR EACH CASE OF
          number of occurrences: 3
               syntactic category: args problem
               conceptual category: none
               status: ok
               priority: na

     t8) Errors due to type ahead problems
          e.g. NOTE FIDO ISA DOG FORGET FIDO LIKES BEER
          number of occurrences: 2
               syntactic category: none
               conceptual category: none
               status: ok
               priority: na
```

BEST COPY AVAILABLE

t11) Over legal limit for number of characters in a node name
e.g. FOR EACH CASE OF RICHARD WISHYWASHYONE ?
number of occurrences: 3
syntactic category: character limit overflow
conceptual category: none
status: different error message
priority: high

t12) Spurious control characters in node or relation names
e.g. FOR EACH CASE OF FIDO ISTA DOG
number of occurrences: 2
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high

t13) Steamroller effect by the synonym and spelling corrector
e.g. 3 FORM /X/ /Y/
number of occurrences: 1
SOLO makes this into FOR and gives the error message)
syntactic category: none
conceptual category: none
status: trapable
priority: high

t14) Over extended use of wild card
e.g. FOR EACH CASE OF ?---LIKES--- BEER
number of occurrences: 7
syntactic category: variable constraint violation
conceptual category: generalized variable positioning validity
status: improved message
priority: medium

t15) Failure to type full procedure name
e.g. FOR FIDO ISA ?
(FOR is used instead of FOR EACH CASE OF)
number of occurrences: 4
syntactic category: not typing out the proper syntax required
conceptual category: truncation legality assumed
status: trapable
priority: low

t26) Forgot procedure name
e.g. SOLO: 6---ismorethan--- 7
number of occurrences: 12
syntactic category: missing procedure name, arguments only
conceptual category: none
status: trapable
priority: medium

t34) "*" /"?" confusion
e.g. FOR EACH CASE OF FIDO ISA *
number of occurrences: 13
syntactic category: variable/value reversal
conceptual category: variable/value concept or name confusion
status: trapable
priority: medium

192

BEST COPY AVAILABLE

t35) FOR EACH CASE OF using only a triple
e.g. FOR EACH CASE OF FIDO ISA DOG
number of occurrences: 32
syntactic category: using a constant for a variable slot
conceptual category: iteration control error
status: improved message
priority: medium

MESSAGE NUMBER 57

Message name: The format for X is: /~X is a user defined procedure~7
parameter 1, parameter 2, parameter 3 (at most 3 parameters)

Summary statistics

Total number of occurrences: 21
Total number of proper occurrences of error message(ok): 0
number of trapable errors: 21
number of errors which need a new or improved error message: 0

Error Types

t26) Forgot to include an obvious SOLO procedure name
e.g. 3 "BETTER LUCK NEXT TIME"
(forgot PRINT)
number of occurrences: 6
syntactic category: missing procedure name, arguments only
conceptual category: none
status: trapable
priority: medium RTC: 2

t75) Addition of sublines (e.g. 2A) where one is not legal
e.g. ...: 2A PRINT "THIS IS THE NEXT STEP"
/~line 2 doesn't require a substep~7
(2A = 2 A) Oops the format for A is:
number of occurrences: 3
syntactic category: subline not legal
conceptual category: subline construct legal in all places or made an error editing
status: trap
priority: high RTC: 4

t32) Spelling error
e.g. 3 P RINT "NOT AVAILABLE"
The format for P is:
number of occurrences: 7
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t8) Errors due to typeahead problems
e.g. 10 10 FIDOBARK /X/ to *A
Oops the format for 10 is:
number of occurrences: 3
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

BEST COPY AVAILABLE

t77) Used 0 instead of Ø or 1 instead of 1
e.g. 20 FORGET FIDO ISA DOG
(20 = 2 0)
Oops... the format for 0 is...
number of occurrences: 2
syntactic category: wrong symbol used
conceptual category: none
status: trap
priority: high RTC: 5

MESSAGE NUMBER 63

Message name: I don't have a HELP option called

Summary statistics

Total number of occurrences: 20
Total number of proper occurrences of error message(ok): 7
number of trapable errors: 13
number of errors which need a new or improved error message: 0

Error Types

t8) Errors due to type ahead problems
e.g. Type option here: TRACE EDIT FOO
number of occurrences: 7
syntactic category: none
conceptual category: none
status: ok
priority: na

t32) Spelling error
e.g. type option here: OSPELL
number of occurrences: 9
syntactic category: mispell
conceptual category: none
status: trapable
priority: medium

t36) Tried to get out of fancy feature
e.g. type option here: DONE or END
number of occurrences: 2
syntactic category: arg not in specified arg set
conceptual category: escape from mode error
status: trapable
priority: medium

t37) Tried to use COPY as a HELP option
e.g. type option here: COPY
number of occurrences: 2
syntactic category: wrong input error
conceptual category: mode/feature access error
status: trapable
priority: low

BEST COPY AVAILABLE

MESSAGE NUMBER 64

Message name: infinite loop (recursion) suspected. Type HELP for advice

---

Summary statistics

Total number of occurrences: 64
Total number of proper occurrences of error message(ok): 64
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t38) Student in infinite loop
e.g. (student simply defines a recursive procedure with improper flow of control)
number of occurrences: 64
syntactic category: none
conceptual category: flow of control error
status: ok
priority: na

MESSAGE NUMBER 65

Message name: O.K.... I now know how to X

Summary statistics

Total number of occurrences: 274
Total number of proper occurrences of error message(ok): 274
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t65) student successfully concludes the creation of a procedure (using TO)
e.g. ....: DONE
ok... I now know how to 'PRAISE' /X/
number of occurrences: 274
syntactic category: none
conceptual category: none
status: ok
priority: na

MESSAGE NUMBER 66

Message name: OK... I have redefined how to X

Summary statistics

Total number of occurrences: 339
Total number of proper occurrences of error message(ok): 339
number of trapable errors: Ø
number of errors which need a new or improved error message: Ø

Error Types

BEST COPY AVAILABLE

t66) Student successfully concludes the editing of a procedure using EDIT
e.g. ...: DONE
OK... I have redefined how to PRAISE /X/
number of occurrences: 339
syntactic category: none
conceptual category: none
status: ok
priority: na

MESSAGE NUMBER 71

Error name: can't copy 'FOR EACH' until you first use it yourself

---

Summary statistics

Total number of occurrences: 2
Total number of proper occurrences of error message(ok): 2
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t39) Attempt to COPY a FOR EACH procedure before state change
e.g. (title is self explanatory)
number of occurrences: 2
syntactic category: none
conceptual category: possibly doesn't understand the change in state of system form wildcard to FOR EACH state
status: ok
priority: na

MESSAGE NUMBER 72

Message name: wrong subline format

Summary statistics

Total number of occurrences: 1
Total number of proper occurrences of error message(ok): 1
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t40) Mixed syntax at edit time
e.g. ..... 10B IF PRESENT:
Oops... wrong subline prompt IF PRESENT for 1B
(should be "10B IF ABSENT:" or "10A IF PRESENT:")
number of occurrences: 1
syntactic category: mixed syntax
conceptual category: possibly inferring to much flexibility to the syntax
status: ok
priority: na

BEST COPY AVAILABLE

MESSAGE NUMBER 76

Message name: (Please specify EXIT, CONTINUE, or NEXT CASE
after the ; below)

Summary statistics

Total number of occurrences: 126
Total number or proper occurrences of error message(ok): 110
number of trapable errors: 12
number of errors which need a new or improved error message: 4

Error Types

t8) Errors due to type ahead problems
e.g. 1A If Present: DONE EDIT FOO
number of occurrences: 3
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

t32) Spelling error
e.g. EXEXIT
number of occurrences: 4
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t67) Missing control flow statement in conditional
e.g. 10A If Present: Print "I can't count higher than this"
(Please specify EXIT, CONTINUE, or NEXT CASE after the
; below)
number of occurrences: 107
syntactic category: missing statement
conceptual category: absent flow of control
status: improve message
priority: low RTC: 1

t71) Student attempts to use NEXT CASE as control flow in a CHECK
statement that isn't within a FOR EACH
e.g. 20A If Present: NEXT CASE
(NEXT CASE ONLY allowed in FOR EACH construct)
number of occurrences: 4
syntactic category: wrong control statement for construct
conceptual category: confusion about concepts and actions
of different constructs
status: improve message
priority: medium RTC: 3

BEST COPY AVAILABLE

MESSAGE NUMBER 81

Message name: Wild card here it now taboo (type help for explanation)

Summary statistics

Total number of occurrences: 21
Total number of proper occurrences of error message(ok): 21
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t41) Use of wild card in check after using the FOR EACH construct
e.g. (title is self explanatory)
number of occurrences: 21
syntactic category: none
conceptual category: not understanding the change in state of the system
status: ok
priority: na

MESSAGE NUMBER 86

Message name: Record for user "X" not found

Summary statistics

Total number of occurrences: 3
Total number of proper occurrences of error message(ok): 3
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t42) User record not found in the data base
e.g. (title is self explanatory)
number of occurrences: 3
syntactic category: possibly mis-typed student number
conceptual category: none
status: ok
priority: na

BEST COPY AVAILABLE

MESSAGE NUMBER 88

Message name: your * variable refers back to more than one 'FOR EACH'

Summary statistics

Total number of occurrences: 4
Total number of proper occurrences of error message(ok): 4
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t43) Non-unique variable names in a global variable situation
e.g. (person uses a "*" variable in more than one FOR EACH statement)
number of occurrences: 4
syntactic category: non-unique variable name referenced
conceptual category: locality of variable problem possibly
status: ok
priority: na

MESSAGE NUMBER 89

Message name: you may only use 'FOR EACH' within a procedure definition

---

Summary statistics

Total number of occurrences: 2
Total number of proper occurrences of error message(ok): 1
number of trapable errors: 1
number of errors which need a new or improved error message: 0

Error Types

t18b) Mode confusion: thought in editor
e.g. SOLO: 2 FOR EACH CASE OF X Y ?
number of occurrences: 1
syntactic category: none
conceptual category: mode error
status: trapable
priority: medium

t44) FOR EACH used at top level
e.g. SOLO: FOR EACH CASE OF X---Y--- ?
number of occurrences: 1
syntactic category: wrong mode for the construct
conceptual category: mode confusion, or misunderstanding constraints of construct
status: ok
priority: na

BEST COPY AVAILABLE

MESSAGE NUMBER 91

Message name: Spelling Checker "When you typed X did you mean Y? (Yes or No):

Summary statistics

Total number of occurrences: 369
Total number of proper occurrences of error message(ok): 179
number of trapable errors: 190
number of errors which need a new or improved error message: 0

Error Types

t85) Spelling checker asks for confirmation on a plausible (word ± 2 chars) spelling change; Accepted by user
e.g. when you typed NAYHAVE did you mean MAYHAVE (yes or no): yes
number of occurrences: 130
syntactic category: mispell
conceptual category: none
status: ok
priority: na

t86) Spelling checker asks for confirmation on a plausible change (word ± 2 chars); Rejected by user
e.g. when you typed JEAN did you mean JOAN
number of occurrences: 49
syntactic category: none
conceptual category: none
status: ok
priority: na

t87) Spelling checker does a silly number check
e.g. When you typed 14 did you mean 1
number of occurrences accepted: 2
number of occurrences rejected: 93
syntactic category: in accepted case above; mispell
conceptual category: none
status: trap
priority: high

t88) Spelling checker tries a ridiculous possible match (word +/->2 chars
e.g. When you typed JUMPROPE did you mean J?
number of occurrences: Accepted: 4
number of occurrences: Rejected: 93
syntactic category: in accepted case above; misspell
conceptual category: none
status: trapable
priority: high

BEST COPY AVAILABLE

MESSAGE NUMBER 97

Message name: Oops... the format for COPY is:
COPY procedure-name (student ID)

Summary statistics

Total number of occurrences: 6
Total number of proper occurrences of error message(ok): 0
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t12) Spurious control characters in node or relation names
e.g. COPY PROBLEM BEGIN GO H0225432
number of occurrences: 1
syntactic category: spelling error or noisy line
conceptual category: none
status: trapable
priority: high RTC: 2

t10) Illegal characters in node or relation names
e.g. COPY CONFIRM FO-223700
number of occurrences: 1
syntactic category: spelling error
conceptual category: none
status: trapable
priority: high RTC: 2

t3) The user adds an unwanted space to the node/relation or procedure name after showing that s/he understands the format
e.g. COPY UPPER NUM DO166843
number of occurrences: 1
syntactic category: spelling error
status: trapable
priority: low RTC: 2

t23a) Gave procedure name with TO attached to it
e.g.COPY TO CONFIRM B002613
number of occurrences: 1
syntactic category: included procedure definition term in procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high RTC: 3

t23b) Gave procedure name with args included to a procedure
e.g. COPY TO CONFIRM /X/ /Y/ /Z/ P0026213
number of occurrences: 2
syntactic category: included args as part of procedure name
conceptual category: procedure argument requirement confusion
status: trapable
priority: high RTC: 3

BEST COPY AVAILABLE

MESSAGE NUMBER 100

Message name: SPELLING CORRECTOR

Summary statistics

Total number of occurrences: 310
Total number of proper occurrences of error message(ok): 272
number of trapable errors: 38
number of errors which need a new or improved error messag.: 0

Error Types

t83) Spelling corrector makes correct change to input line
e.g. "10PRINT = 10 PRINT"
number of occurrences: 271
syntactic category: missing space delimiter
conceptual category: none
status: ok
priority: na

t75) Split line number because of addition of inappropriate subline
e.g. 2A FORGET = 2 A FORGET
number of occurrences: 21
syntactic category: subline not legal
conceptual category: subline construct legal in all places or made an error editing
status: trap
priority: High RTC: 4

t13) Steamroller effect by the synonym and spelling corrector
e.g. COMPARE = CHECK
Oops... the format for CHECK is:
number of occurrences: 9
syntactic category: none
conceptual category: none
status: trapable
priority: high RTC: 5

t80) User uses a synoym for a SOLO procedure for which no synonym exists in the spelling corrector
e.g. DELETE = FORGET
(SOLO substitutes desired synonym)
number of occurrences: 1
syntactic category: illegal procedure name
conceptual category: use of synonym instead of legal procedure name
status: ok
priority: na

t84) Spell corrector gives the wrong split to a user input line
e.g. POSSIBLYHAS = PO SSIBLYHAS
or
10ASSIGN = 10A SSIGN
number of occurrences: 8
syntactic category: missing space
conceptual category: none
status: trapable
priority: medium

BEST COPY AVAILABLE

MESSAGE NUMBER 101

Message name: Type Option here:

Summary statistics

Total number of occurrences: 50
Total number of proper occurrences of error message(ok): 25
number of trapable errors: 25
number of errors which need a new or improved error message: 0

Error Types

t63) No response to an input mode prompt
e.g. Type option here:
(no response, only carriage return)
number of occurrences: 21
syntactic category: no arg given
conceptual category: not understanding error
status: improve message
priority: high RTC: 3

t32) Spelling error
e.g. Type option here: OSILENCE
number of occurrences: 17
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t17) Wrong access to the help features
e.g. Type option here: TRACE RECOGNISE
number of occurrences: -
syntactic category: wrong mode for input
conceptual category: help mode accessible directly
from SOLO top level
status: trapable
priority: medium RTC: 3

t8) Errors due to type ahead problems
e.g. Type Option here: EDIT FOO
number of occurrences: 4
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

t1) Too many arguments to a procedure
e.g. Type option here: NO SILENCE
or
SILENCE (-)
number of occurrences: 6
syntactic category: too many args
conceptual category: unlimited lists possible
status: improve message
priority: low RTC: 2

BEST COPY AVAILABLE

Summary sheet for:

101) 'Help Options' e.g. "Type Option here:"

Feature	number of occurrences
SPELL	2
NOSPELL	16
TRUSTING	Ø
CAUTIOUS	Ø
NODEON	Ø
NODEOFF	Ø
REACT	6
SILENCE	111
TRACE	65
NOTRACE	33
BRIEF	5
VERBOSE	1
RESTORE	2
START	Ø
REVERT	16
CAT	132

Total number of occurrences:	389
Total number of error message:	25
Total number of trapable:	25
TOTAL	439

BEST COPY AVAILABLE

MESSAGE NUMBER 102

Message name: Oops... A---A--- A already

Summary statistics

Total number of occurrences: 183
Total number of proper occurrences of error message(ok): 183
number of trapable errors: 0
number of errors which need a new or improved error message: 0

Error Types

t76) Attempts multiple nodes from a single node-relation pair
While in "CHECK mode"
e.g. Note 2ADD---6--- 8
Oops... 2ADD---6--- 9 already
number of occurrences: 183
syntactic category: none
conceptual category: overwriting possible or confusion about uniqueness of triples
status: improve
priority: low RTC: 2

MESSAGE NUMBER 113

Message name: Do you want to select any more options?

Summary statistics

Total number of occurrences: 573
Total number of proper occurrences of error message(ok): 570
number of trapable errors: 3
number of errors which need a new or improved error message: 0

Error Types

t50) Positive response to a confirmation statement
e.g. Do you want to select any more options? YES
number of occurrences: 129
syntactic category: none
conceptual category: none
status: ok
priority: na
RTC: 2

t8) Errors due to type ahead problems
e.g. Do you want to select any more options? EDIT FOO
number of occurrences: 19
syntactic category: none
conceptual category: none
status: partially trapable RTC: 3

BEST COPY AVAILABLE

t32) Spelling error
e.g. Do you want to select any more options? ON
(meant to spell no)
number of occurrences: 3
syntactic category: misspell
conceptual category: none
status: trapable
priority: medium RTC: 2

t7) No arguments given
e.g. Do you want to select any more options?
number of occurrences: 34
syntactic category: args problem
conceptual category: none
status: ok
priority: na

t51) Negative response to a confirmation statement
e.g. Do you want to select any more options?: NO
number of occurrences: 388
syntactic category: none
conceptual category: none
status: ok
priority: na RTC: 2

BEST COPY AVAILABLE

COMPUTER-ASSISTED LEARNING RESEARCH GROUP

THE OPEN UNIVERSITY

List of Technical Reports

These reports may be obtained (free of charge but subject to availability) from:-

Ann Jones,
Institute of Educational Technology,
The Open University,
Walton Hall,
MILTON KEYNES, MK7 6AA,
England.

Report No.	Title and Author
1	M.A. Bramer (1980) Using Computers in Distance Education: The first ten years of the British Open University.
3	M.A. Bramer (1980) COMAL-80 - Adding Structure to BASIC.
4	A. Jones (1980) An evaluation of the SOLO programming language: A pilot study of its use at summer school.
5	A. Jones, T. O'Shea (1981) An evaluation of tutorial CAL at the Open University: the use of Cicero in SP7286.
6	E.L. Whitelegg (1981) An evaluation of tutorial CAL at the Open University: the use of MERLIN in S7291.
7	M.W. Lewis (1980) Improving SOLO's User-Interface: An Empirical study of user behaviour and a proposal for cost-effective enhancements to SOLO.
8	A.T. Vincent (1980) Some applications of a microcomputer.
9	T. O'Shea, A Floyd (1981) Recording Children's mathematical behaviour
10	A.T. Vincent (1981) Computer-assisted support for blind students: the use of a microcomputer-linked voice synthesizer.
11	P. Bacsich (1981) Viewdata-style delivery mechanisms for CAL.
12	P.J. Murphy (1981) Biological simulations in distance learning.
13	P.J. Murphy (1981) Computer simulation in Education: Genetics and Evolution.
14	I. Every (1981) Graphics and animation in teaching dialogues.
15	P.J. Murphy, E. Scanlon, E.L. Whitelegg (1981) An evaluation of simulation CAL in the Science Faculty during 1980.
16	M. Eisenstadt, J. Laubsch, J.H. Kahney (1981) Creating pleasant programming environments for cognitive science students.
17	J. Laubsch, M. Eisenstadt (1981) Domain specific debugging aids for novice programmers.
18	J. Roberts (1981) Are fuzzy grids really necessary?
19	J. Preece (1981) Investigating how students interpret cartesian graphs.
20	E. Scanlon, Tom Vincent (1981) CAL in Science: Some Regional Implications.
21	A. Jones (1981) The representation of conceptual models: some issues.
22	E. Scanlon (1982) Improving Problem Solving in Physics.
23	M. Sharples (1982) CYCLOPS - A case study in the design of user-friendly educational technology.

BEST COPY AVAILABLE